Synchronization and Channel Estimation in OFDM: Algorithms for Efficient Implementation of WLAN Systems

Von der Fakultät für Mathematik, Naturwissenschaften und Informatik
der Brandenburgischen Technischen Universität Cottbus

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
genehmigte Dissertation
vorgelegt von

Diplom Ingenieur

Alfonso Luís Troya Chinchilla

geboren am 14. Dezember 1975 in L'Hospitalet del Llobregat (Spanien)

Gutachter: Prof. Dr.-Ing. Rolf Kraemer

Gutachter: Prof. Petri Mähönen

Gutachter: Prof. Dr.-Ing. Klaus-Rüdiger Fellbaum

Tag der mündlichen Prüfung: 12. Juli 2004

Verlange nicht, dass alles, was geschieht, so geschieht, wie du es willst,
sondern wünsche dir, dass alles so geschieht, wie es geschieht,
und du wirst glücklich sein.

Epiktet (geb. um 50 n. Chr., ges. um 140 n. Chr.), „Handbuch der Moral“

Preface

THE WORK FOR THE PRESENT DISSERTATION has been carried out at the IHP (Institute for High Performance microelectronics) in Frankfurt an der Oder, Germany, as part of the so-called Wireless Broadband Networks (WBN) Project under supervision of Professor Rolf Kraemer and is to be submitted to the Faculty of Informatics at the Brandenburgische Technische Universität (BTU) Cottbus. I would like to thank Professor Kraemer for the confidence he put on me when starting this Project in 1999, providing me the possibility to carry out a Ph.D. without being directly involved at the University. The author would also like to thank all the members of the WBN Project for their support and in particular Dr. Eckhard Grass for his great work as project leader, Dr. Gunther Lippert and Mr. Ulrich Jagdhold for their support in the difficult initial moments.

I want to express my sincere gratitude to Dr. Koushik Maharatna and Mr. Miloš Krstić for their contributions in this Dissertation. Several of the proposals given in this work have materialized thanks to their experience in digital design. Also, thank you very much indeed for your contributions, discussions and corrections to all our co-authored papers. Beyond the call of duty, Koushik has been a great friend with whom the tough days commuting from Berlin to Frankfurt became more pleasant.

I would like to give my warmest thanks to my parents, Ildefonso Troya Márquez and María Dolores Chinchilla López for their moral support while staying far away from home.

This work is dedicated to all the friends from several nationalities I have met in Frankfurt (Oder), and specially to Nina. Our Souls joined each other in Frankfurt, and so they will remain forever.

Frankfurt an der Oder / Berlin, January 2004

Abstract

SEVERAL STANDARDS FOCUSED ON WIRELESS COMMUNICATION SYSTEMS have been released in the last few years. They have to be distinguished from the Mobile communication systems, e.g. GSM or UMTS, in the sense that they require much less mobility at the expense of higher transmission rates, as it is the case of wireless MANs, LANs, PANs, (Metropolitan, Local and Personal Area Networks, respectively) and Home Networks. Among them we can mention Bluetooth™, OpenAir and HomeRF, as purely industry-generated standards. Furthermore, we find the American IEEE standards of the 802.11 group (WLANs), the 802.15 group (WPANs) and the 802.16 group (WMANs). These standards find a European equivalence in the ETSI Project BRAN with the HIPERLAN, HIPERACCESS and HIPERLINK standards.

A key point to be considered when implementing wireless systems is their power consumption. These systems are generally installed in mobile phones, PDAs or Laptop computers, which in general have a limited power source. Therefore, efficient designs demanding low power are a principal focus of investigation.

In addition to this, the OFDM transmission scheme is gaining interest in the most recent standard definitions due to its high spectral efficiency and ability to overcome multipath fading. However, OFDM is very sensitive to impairments caused by the analog transmission, hence requiring powerful correction algorithms.

This Dissertation is a contribution to the design of the Synchronization and Channel Estimation algorithms in Wireless OFDM systems, paying special attention to their implementation. After investigation of the main impairments affecting OFDM in a wireless transmission, the Dissertation obtains solutions for all the blocks forming the so-called Inner Receiver. The IEEE 802.11a standard is taken in this work as a reference, since this is the first standard proposal in which OFDM is applied for wireless LAN with transmission rates of up to 54 Mbps. The low-power feature of our proposals has been demonstrated by designing an Integrated Circuit fully compatible with the IEEE 802.11a specifications. Results show that the power figures expected for our design are very competitive in comparison with the results reported by other research groups and companies working in this field.

Keywords: 802.11a, Channel Estimation, HiperLAN2, OFDM, Synchronization, WLAN.

Deutsche Kurzfassung

In den letzten Jahren ist eine ganze Reihe von WLAN Standards im Bereich der drahtlosen Kommunikationssysteme eingeführt worden. Im Gegensatz zu GSM oder UMTS ist die Mobilität dieser Systeme sehr beschränkt. Auf der anderen Seite werden sehr hohe Übertragungsraten erreicht. Ein Beispiel dafür sind die drahtlosen MANs, LANs, PANs (jeweils Metropolitan, Local und Personal Area Networks) und die sogenannten Home Networks. Unter diesen befinden sich industrie-unterstützte Standards wie Bluetooth™, OpenAir und HomeRF. Darüber hinaus unterstützt das amerikanische Standardisierungsgremium IEEE die Standards 802.11 (WLAN), 802.15 (WPAN) und 802.16 (WMAN). Diese Standards finden eine europäische Äquivalenz jeweils mit den Standards HIPERLAN, HIPERACCESS und HIPERLINK, die von ETSI unterstützt werden.

Ein wichtiger Punkt, den man in Sinne der Implementierung drahtloser Kommunikationssysteme berücksichtigen muss, ist der Leistungsverbrauch solcher Systeme. Sie werden üblicherweise in Mobiltelefonen, persönlichen Assistenten (PDA) oder Laptops installiert, welche über eine sehr begrenzte Energiequelle verfügen. Aus diesem Grund ist die Entwicklung energieeffizienter Lösungen zu einem äußert wichtigen Forschungsschwerpunkt geworden.

Des Weiteren ist die OFDM Übertragungstechnik zu einer wichtigen Technik in den neuesten Standards dieses Bereiches geworden. Zwei Gründe sprechen für diese Tendenz: erstens, weil OFDM spektral sehr effizient ist; und zweitens, weil OFDM es erlaubt, die Effekte der Mehrwegeausbreitung beim Empfänger in einer deutlich einfacheren Weise als in den herkömmlichen Kanalentzerrern zu kompensieren. Jedoch ist OFDM besonders empfindlich gegenüber den Beeinträchtigungen, die von den analogen Schaltungsblöcken und dem Übertragungskanal verursacht werden, so dass leistungsfähige Korrekturalgorithmen eingesetzt werden müssen.

Diese Dissertation ist ein Beitrag zu der Entwicklung von Synchronisations- und Kanalschätzungsalgorithmen im Bereich der drahtlosen OFDM-Systeme, insbesondere im Hinblick auf ihre Implementierung. Nach einer detaillierten Untersuchung aller Beeinträchtigungen eines OFDM-Signals im Falle einer drahtlosen Übertragung, stellt die Dissertation Lösungen für die verschiedenen Bestandteile des sogenannten „Inner Receiver“ vor. Der Standard IEEE 802.11a wird in dieser Arbeit als Referenz genommen, da erstmalig OFDM als Übertragungstechnik in einem drahtlosen lokalen Netzwerksystem mit Übertragungsraten bis zu 54 Mbps eingesetzt wird. Der niedrige Energieverbrauch unserer Lösung wird durch die Entwicklung eines Chips (Integrierte Schaltung), welcher völlig kompatibel mit dem 802.11a Standard ist, demonstriert. Die Ergebnisse zeigen, dass die hier entwickelte Lösung konkurrenzfähig ist, und einem Vergleich mit publizierten Parametern anderer Entwickler standhält.

Schlagwörter: 802.11a, Kanalschätzung, HiperLAN2, OFDM, Synchronisation, WLAN.

Contents

Abbreviations

AAF	Anti-Alias Filter
ACF	Autocorrelation Function
ADC	Analog-to-Digital Converter
AFE	Analog Front-End
AGC	Automatic Gain Control
AP	Access Point
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BAN	Body Area Network
BBP	Baseband Processor
BCH	Broadcast Channel
BPSK	Binary Phase Shift Keying
BRAN	Broadband Radio Access Network
CCDF	Complementary Cumulative Distribution Function
CD3	Coded Decision-Directed Demodulation
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
CORDIC	COordinate Rotation DIgital Computer
CP	Cyclic Prefix
CPE	Common Phase Error
CRLB	Cramer-Rao Lower Bound
CTF	Channel Transfer Function
DA	Data-Aided
DAB	Digital Audio Broadcasting
DAC	Digital-to-Analog Converter

DD	Decision-Directed
DFT	Discrete Fourier Transform
DLC	Data Link Control layer
DLCP	DLC Processor
DMT	Discrete Multi-Tone
DPLL	Digital Phase-Locked Loop
DRM	Digital Radio Mondiale
DTL	Digital Timing Loop
DVB	Digital Video Broadcasting
EPP	Enhanced Parallel Port
ETSI	European Telecommunication Standards Institute
EVM	Error Vector Magnitude
FAP	False Alarm Probability
FEC	Forward Error Correction
FEQ	Frequency Equalization
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GCA	Gain Controlled Amplifier
HL2	HiperLAN2
IBO	Input Back-off
ICI	Inter-Carrier Interference
IDFT	Inverse Discrete Fourier Transform
IEEE	Institute of Electrical and Electronic Engineers
IF	Intermediate Frequency
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
ISI	Inter-Symbol Interference
ISM	Industrial, Service and Medical spectral band
ISR	Interference-to-Signal Ratio
LO	Local Oscillator
LOS	Line Of Sight
LS	Least Squares
MAC	Medium Access Control
Mbps	Mega bits per second
MCM	Multi-Carrier Modulation
ML	Maximum Likelihood
MMSE	Minimum Mean-Squared Error
MSE	Mean-Squared Error
MT	Mobile Terminal
NCO	Numerically Controlled Oscillator

NDA	Non Data-Aided
NLOS	No Line Of Sight
NRM	Noise Reduction Matrix
OFDM	Orthogonal Frequency Division Multiplexing
PAN	Personal Area Network
PAPR	Peak-to-Average Power Ratio
PDA	Personal Data Assistant
PDF/pdf	Probability Density Function
PDP	Power Delay Profile
PDS	Power Density Spectrum
PHY	Physical layer
PLL	Phase-Locked Loop
PN	Pseudo-Noise sequence
PTS	Partial Transmit Sequence
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RPE	Residual Phase Error
RRC	Root-Raised Cosine
SiGe:C	Silicon-Germanium-Carbon technology process
SM	Selecting Mapping
SNR	Signal-to-Noise Ratio
SSPA	Solid-State Power Amplifier
TEP	Timing Error Probability
TSM	Transmit Spectral Mask
UMTS	Universal Mobile Telecommunication System
VLSI	Very Large Scale Integration
WBAN	Wireless Body Area Network
WIGWAM	WIreless Gigabit With Advanced Multimedia support
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WSS	Wide-Sense Stationary

Symbols and Notation

$\otimes$	Cyclic convolution
$*$	Normal convolution
$\mathcal{F}_{\tau}\{.\}$	Fourier Transform with respect to the variable τ
$\Re\mathrm{e}\{.\}$	Real part of complex number
$\Im\mathrm{m}\{.\}$	Imaginary part of complex number
$E\{.\}$	Statistical expectation
$(.)^{*}$	Conjugated value
$(.)^{\mathbf{H}}$	Transposed and conjugated version of a matrix
$\lfloor x \rfloor$	Floor operation, i.e. largest integer smaller than or equal to the real value x
$A_{k,l}$	Transmitted value on sub-carrier k in symbol l
$\tilde{A}_{k,l}$	Estimated value on sub-carrier k in symbol l
b	Generic number of bits
$\mathrm{B_c}$	Coherence bandwidth
$\mathrm{B_d}$	Doppler spread
$\mathrm{B_W}$	Total signal bandwidth
$c_{\mathrm{REF}}(m)$	Reference crosscorrelation signal
$C_{k,l}$	Transmitted value of pilot on sub-carrier k in symbol l
e_{ε}	ML frequency error discriminator
e_{θ}	ML time error discriminator
$\mathrm{e_C}(k,l)$	Constellation error
f_{C}	RF carrier frequency
f_{ε}	Carrier frequency offset
f_{IF}	Intermediate frequency
f_{LO}	Frequency of the local oscillator

f_S	Sampling frequency
$\mathbf{F}$	IDFT matrix
$\mathbf{F}^{\mathbf{H}}$	DFT matrix
$\phi_h(\tau)$	Power delay profile
$h(t,\tau)$	Channel impulse response
$\hat{\mathbf{h}}_l$	Estimation of channel impulse response for symbol l
$h_I(i + \mu)$	Impulse response of the variable interpolation filter
$h_i(\mu)$	Impulse response of the variable interpolation filter
$H(t,f)$	Channel transfer function
$H_A(z)$	All-pass filter
$H_{DEC}(z)$	Decimation filter
$H_{INT}(z)$	Interpolation filter
$H_{k,l}$	CTF coefficient for sub-carrier k in symbol l
$H_{k,REF}$	Reference channel transfer function
$\hat{H}_{k',l'}$	Least-Square channel estimation for pilot at sub-carrier k' in symbol l'
$\tilde{H}_{k,l}$	Final channel estimation for sub-carrier k in symbol l
$\lvert H(\Omega, e^{j\omega})\rvert$	Magnitude of filter with zero attenuation at pulsations given by vector Ω
$J(\theta)$ / $J(k)$	Autocorrelation function (general)
$J_C(k)$	Autocorrelation function used to obtain the coarse frequency offset
$J_{diff}(k)$	Differentiated version of $\lvert J_F(k)\rvert^2$
$J_F(k)$	Autocorrelation function used to obtain the fine frequency offset
k	Discrete frequency variable (unless otherwise noted)
K	Length of the convolutional encoder
K_Φ	Total phase noise power
l	Symbol index (unless otherwise noted)
L	Delay introduced by the feedback loop in the channel estimator
$L(\mathbf{r};\boldsymbol{\theta})$	Maximum-Likelihood function
$L_{-26, +26}$	Long preamble symbol in the frequency domain (802.11a / HiperLAN2)
m_1	Position of first peak in $\lvert r_Y(m)\rvert^2$
m_2	Position of second peak in $\lvert r_Y(m)\rvert^2$
m_k	Basepoint in the variable interpolator
M_I	Samples per symbol used by the error detector in the DTL
n	Discrete time variable (unless otherwise noted)
N	Total number of sub-carriers in an OFDM system
N_{acc}	Number of accumulated samples in the group peak detector
N_{avg}	Length of the moving average inside the autocorrelator
N_d	Delay inside the autocorrelator
N_F	Ideal separation of pilots in the frequency domain (if used for interpolation)
N_G	Cyclic prefix length (in samples)

$N_{INT/DEC}$	Interpolation/Decimation factor
N_L	Ideal separation of pilots in the time domain (if used for interpolation)
N_m	Maximum excess delay (in samples)
N_{soft}	Number of soft-bits used during decoding
N_{tap}	Cardinality of $\mathcal{T}_{k,l}$
N_u	Number of used sub-carriers in an OFDM system
N_v	Number of virtual sub-carriers in an OFDM system
P	Number of paths in a multipath channel
$P_{k,l}$	Received value of pilot on sub-carrier k and symbol l
$P_{k,l}^{\phi}$	Residual phase error for pilot k in symbol l
$\mathcal{P}$	Whole set of pilot sub-carriers available in one frame
$\|\mathcal{P}\|$	Cardinality of $\mathcal{P}$
$P_R(\Psi)$	Polynomial in the Laplace domain
$r(t)$	Received OFDM signal
$\mathbf{r}$	Observation vector of received samples
$r_I(t)$	Real part of $r(t)$
$r_{n,l}$	Discrete time domain received OFDM signal
$r_Q(t)$	Imaginary part of $r(t)$
$r_Y(m)$	Crosscorrelator output
R	Degrees of freedom in the design of the linear-phase IIR filters
$R(l)$	Another definition of the autocorrelation function
$R_{\psi N}(l)$	Discrete autocorrelation function of the phase noise
$R_m(\theta)$	Matrix describing plane rotations
$s(t)$	Transmitted OFDM signal
$\mathbf{s}$	Observation vector of transmitted samples
$s_I(t)$	Real part of $s(t)$
$s_l(t)$	Transmitted OFDM symbol l
$s_{n,l}$	Discrete time domain transmitted OFDM signal
$s_Q(t)$	Imaginary part of $s(t)$
$S(\Delta t, \delta)$	Pilot-based time error discriminator
$S_{-26,+26}$	Short preamble symbols in the frequency domain (802.11a / HiperLAN2)
$SA_{-26,+26}$	Short preamble symbols in the frequency domain (HiperLAN2 only)
$S_H(\lambda)$	Doppler power spectrum
t	Time variable (continuous)
t_θ	Timing offset (continuous)
T	Duration of OFDM symbol (excluding cyclic prefix)
T_d	Coherence time
T_G	Duration of the cyclic prefix (or guard interval)

T_I	Ideal sampling interval
T_m	Maximum excess delay
T_S	Sampling interval at the receiver side
T_{SYM}	Duration of OFDM symbol (including cyclic prefix)
$\mathcal{T}_{k,l}$	Sub-set of pilots assigned to sub-carrier k in symbol l
$w(t)$	Conforming pulse in the time domain
$w_{k,l,k',l'}$	Coefficients of the Wiener interpolator
$w_F(k-k')$	Coefficient of the one-dimensional frequency direction Wiener filter
$w_L(l-l')$	Coefficient of the one-dimensional time direction Wiener filter
$\mathbf{W}$	Rank reduction matrix
$Z_{m,l}$	Received OFDM information on sub-carrier m in symbol l
α	Normalized fine frequency offset
α_i	Fundamental angle rotations in the CORDIC processor
α_r	Roll-off factor
β	Normalized coarse frequency offset
β_Φ	Phase noise bandwidth
Δf	Sub-carrier spacing in an OFDM system
ε	Total normalized frequency offset
$\phi_l(k)$	Residual phase error in sub-carrier k of symbol l
$\Phi_N(f)$	Power density spectrum of phase noise
$\mathbf{\Phi}$	Pilot autocorrelation matrix
γ_I^R	Amplitude mismatch of the I-path in the receiver side
γ_I^T	Amplitude mismatch of the I-path in the transmitter side
γ_Q^R	Amplitude mismatch of the Q-path in the receiver side
γ_Q^T	Amplitude mismatch of the Q-path in the transmitter side
η	Degree of non-stationarity
$\varphi_N(t)$	Phase noise process
φ_R	Phase error during I/Q generation in the receiver side
φ_T	Phase error during I/Q generation in the transmitter side
κ	Forgetting factor
λ	Doppler variable (unless otherwise noted)
$\Lambda(\theta, \varepsilon)$	Log-Likelihood function
μ_k	Fractional delay in the variable interpolator
θ	Normalized timing offset
$\boldsymbol{\theta}$	Estimation parameter vector
$\boldsymbol{\theta}_{k,l}$	Correlation vector
$\mathbf{\Theta_N}$	Noise reduction matrix
ρ_S	Sampling density

$\sigma_{\mathrm{ICI}}{}^2$	Power of Inter-Carrier Interference
$\sigma_{\mathrm{ISI}}{}^2$	Power of Inter-Symbol Interference
τ	Delay variable (continuous)
τ_i	Delay of path *i* in a multipath channel
τ_{rms}	Root-mean square channel delay
$\tau(\omega)$	Group delay
ω_{P}	Maximum value of pulsation in the pass-band
ω_{S}	Minimum value of pulsation in the stop-band
Ω	Vector of pulsations
$\Omega^{(i)}$	Vector of pulsations at iteration *i*
Ψ	Laplace domain variable
$\Psi_n(t)$	OFDM base signal
ζ	Sampling clock frequency offset (in ppm)

Chapter 1

Introduction

1.1 Motivation of this Dissertation

THERE IS AN INCREASING AMOUNT OF DISCUSSION in the scientific community related to fourth-generation (4G) wireless systems that are expected to emerge quite soon after the deployment of third-generation cellular mobile systems. Future 4G networks will be formed by heterogeneous multi-technology systems, where different networks must be able to interoperate and should be designed to be “polite” to each other, e.g. they should cause as low interference as possible to other wireless devices. In this context, it is highly possible that the future 4G infrastructure is not built just around macrocellular, e.g. UMTS, and microcellular, e.g. wireless LAN technologies, but personal and body area networks (PANs and BANs) will also become very important. The general situation is depicted in Figure 1.1, where the most significant mobile communication standards are represented in terms of mobility and achieved data rates.

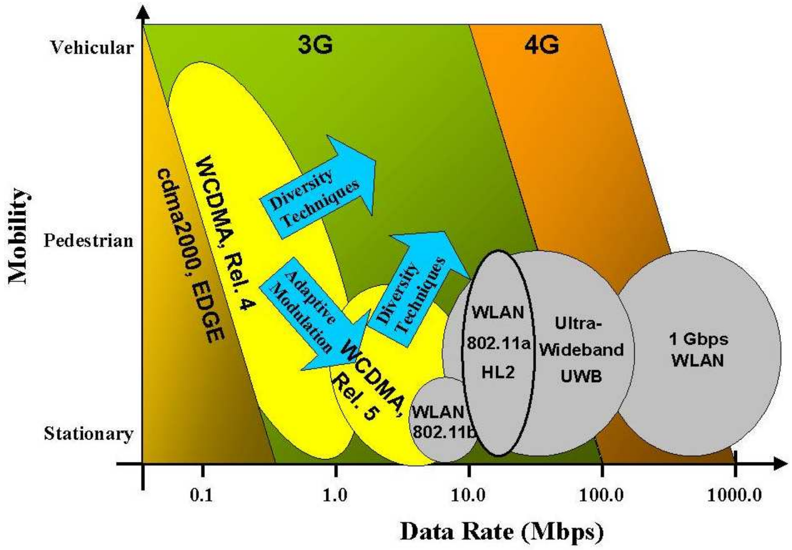


Figure 1.1. Expected evolution of wireless communication systems in terms of data rate and mobility.

Considering the 4G scenario, efficient transmission schemes will play a key role in the achievement of the expected high data rates. One method, which has gained interest in the last years and appears to be a good candidate, is the OFDM (Orthogonal Frequency Division Multiplex) transmission scheme. This method was already presented in the 50's and 60's as an *efficient parallel transmission system*. Nevertheless, its implementation showed a number of technical challenges, which were not possible to solve with the technology available at that time. As an example, an efficient method for digital calculation of the Fourier Transformation, which is the basis of OFDM, was not available until 1965.

With the invention of the FFT algorithm by Cooley and Tukey in 1965 [Cool65] and its efficient implementation on chip, OFDM has become a reality. Presently it has been proposed to be the transmission scheme in the next generation of broadcasting systems, i.e. DAB (Digital Audio Broadcasting), DVB (Digital Video Broadcasting) and DRM (Digital Radio Mondiale), in Power-Line systems and in xDSL (Digital Subscriber Line). However, in all these systems power consumption is not the most restrictive parameter to be concerned about during design, since they have access to a virtually unlimited power source.

The IEEE 802.11a [IEEE] and the ETSI BRAN HiperLAN2 [ETSIa] standards are the first ones defining an OFDM scheme for wireless communications in Local Area Networks (LANs). They are directed to wireless digital burst transmissions up to 54 Mbps between portable devices, which in general have a very limited battery performance. But these standards represent only the tip of the iceberg. As an example, inside the 802.15.3a task group [IEEE15] intensive discussions are going on regarding the application of OFDM or CDMA into the next generation of high-rate WPANs (up to 400 Mbps). The OFDM alternative shows in this case the best chances to prevail because of the strong industrial partners supporting this solution [Bla03]. Yet another example is the WIGWAM Project (Wireless Gigabit with Advanced Multimedia Support) led by the Technical University of Dresden and other partners and to be started late 2003. This project targets data rates of up to 1 Gbps (using most probably OFDM) by the year 2007. How to achieve such high data rates with low power consuming devices is the major scientific challenge.

In order to be able to test some of the 4G networking issues with present-day technology and to provide an evolutionary path toward new solutions, the Wireless Communication Systems Department at the IHP[1] decided to start the WBN Project (Wireless Broadband Networks) in 1999. The aim of this project is the integration into a single chip of both, the PHY (Physical) as well as the DLC (Data Link Control) layers of a modem compliant with the IEEE 802.11a and ETSI BRAN HiperLAN2 standards [Grass01]. Two main challenges arose at that time:

- Due to the huge computation power required by both layers, it had to be decided where to establish the so-called Hardware-Software partitioning, i.e. up to which extent it was possible to realize a solution based on commercially available DSPs.

---

[1] Innovations for High Performance microelectronics, Frankfurt (Oder), Germany.

- In the 802.11a and HL2 standards there is a full description of both DLC and PHY layers. Nevertheless, there is nothing mentioned about the *Acquisition* and *Channel Equalization* strategies to be performed during reception of the OFDM frames.

Regarding the first point, a pure ASIC solution was taken as most appropriate for this purpose. In [Grass01], two main reasons are given to support this decision. On one hand, both standards allow very little latency during digital processing of the baseband signal (on the order of 10 µs) in order to meet the timing constraints for sending acknowledgment frames, which is extremely hard to achieve with a DSP-based solution. On the other side, a pure software solution typically requires at least one order of magnitude more power than a pure dedicated hardware [Xant99]. In our particular design for the 802.11a and HL2 standards, the targeted power consumption for the whole chip is 1 W, including the Analog Front-End (AFE), the Baseband Processor (BBP) as well as the DLC Processor (DLCP), aiming to an architecture suitable for portable devices.

Concerning the second point, acquisition and channel equalization are key subjects when dealing with the PHY layer of OFDM systems. By *Acquisition* (or *Synchronization*) we understand all the algorithms and procedures necessary to detect the OFDM signal at the receiver side, and to estimate and correct the time and frequency offsets. The *Channel Equalization* is necessary to correct for the filtering suffered by the transmitted signal not only due to the transmission channel itself, but also due to the different filters found in the AFEs. All these procedures are embedded into the so-called **Inner Receiver**, as defined in [Meyr97], and shown in Figure 1.2. After coding and modulating the input bits, a complex baseband signal with the form $I_{IN}+jQ_{IN}$ (In-phase and Quadrature components, respectively) is provided to the OFDM transmitter. The RF up-converter makes use of the baseband signal to generate a passband signal by frequency translation and delivers this signal to the antenna. After transmission through the radio channel, the signal reaches the receive antenna and is down-converted back to the baseband. The resulting complex baseband signal will be affected by a number of impairments. Hence, the task of the Inner Receiver is to estimate and compensate all these impairments, aiming to provide a complex baseband signal $I_{OUT}+jQ_{OUT}$ as similar as possible to $I_{IN}+jQ_{IN}$. As it will be seen throughout this Dissertation, the algorithms applied by the Inner Receiver depend very much on the transmission method, either continuous or bursty, and on the availability of training signals. Hence, several approaches may be used for Synchronization and Channel Estimation in OFDM. Our challenge is therefore the derivation of near-optimum and implementation-friendly algorithms, especially suitable for wireless LAN applications, resulting in low-power low-latency realizations.

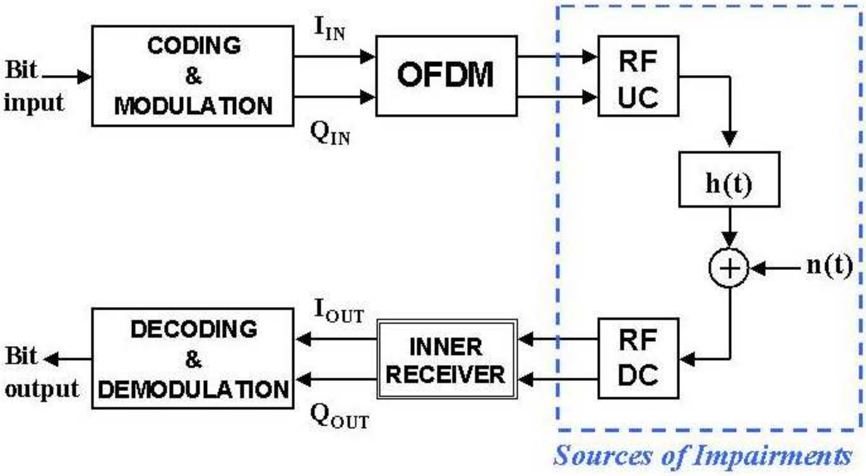


Figure 1.2. Reference system model as proposed in [Meyr97].

1.2 Goals of this Dissertation

Based on our firm believe that OFDM will play a key role in emerging 4G wireless communication systems, this Dissertation has one main objective: obtain practical and affordable designs for the synchronization and channel estimation tasks in OFDM-based wireless LAN, PAN and BAN systems. The IEEE 802.11a standard is taken here as a reference in a non interference-limited scenario. Therefore, this Dissertation will not deal with techniques aiming to mitigate network-induced interferences such as co-channel interferences or interferences derived from co-existing wireless systems. In addition, only the design of the Mobile Terminal (MT) is envisioned, since it is the segment requiring lowest power consumption. To reach these objectives, in particular, the following goals have been pursued:

- Realization of simulation models of the whole transmission chain based on SPW[1] and Matlab[2].
- Detailed study of all the impairments affecting an OFDM transmission scheme in a wireless scenario and establishment of the tolerable limits for these impairments.
- Thorough analysis of existing synchronization methods for OFDM and determination of the most suitable ones for wireless LAN transmissions.
- Examination of existing channel estimation approaches for OFDM and derivation of a suitable architecture for the special case of the 802.11a and HL2 standards.
- Design of a low-power low-latency Inner Receiver for an IEEE 802.11a standard compliant modem according to the previous investigations.
- On-silicon implementation of an experimental baseband processor for the IEEE 802.11a standard.

---

[1] The CoWare™ Signal Processing WorkSystem is a registered trademark of CoWare, Inc.

[2] Matlab® is a registered trademark of The MathWorks, Inc.

1.3 Contents and Important Results

The Dissertation consists of eight chapters and two appendices grouped into two parts. Part I comprises Chapters 2, 3, 4 and 5, and is devoted to the accomplishment of the first four goals listed in §1.2. Thus, Chapter 2 initially introduces a channel model for wireless communications suitable for simulation. From this channel model some meaningful parameters are defined. The OFDM transmission scheme is afterwards mathematically described in detail, and its advantages compared to the traditional single carrier transmission schemes are pointed out. Mainly, an OFDM system can cope with fading channels in a better way than single carrier systems do if it is designed appropriately. The chapter finishes with a detailed explanation of the PHY layer for the 802.11a and HL2 standards, as these will be the reference standards used for the implementation described in Part II of this Dissertation.

Chapter 3 starts with a detailed investigation of the impairments affecting the OFDM signals following the model shown in Figure 1.2. Here, the impact of carrier frequency offsets, timing offsets, sampling clock frequency offsets, phase noise, I/Q imbalances and non-linear distortions is analyzed. It is seen that timing and frequency offsets are the impairments that to a greater extent negatively influence the demodulation and decoding of OFDM signals.

An analysis of the several synchronization algorithms reported in the literature is given in Chapter 4. Synchronization is limited to time and frequency acquisition. The analysis starts with the so-called Blind or Non-Data Aided (NDA) schemes, in which a Maximum Likelihood (ML) approach is used. Although asymptotically optimal, these schemes are seen to have a long acquisition time together with a short acquisition range, making them of little use in practical wireless LAN systems. On the contrary, Data-Aided (DA) algorithms seem to be more suited for the burst transmissions typically found in WLAN. In this context, the synchronization problem is divided into coarse and fine acquisition, and solutions for both are derived. In the special case of timing synchronization, it is further seen that the sampling clock frequency offset is equivalent to a slowly varying timing offset, making it necessary to use tracking algorithms. A suitable pilot-based DA timing discriminator is suggested for this purpose, whose analysis is further developed in Chapter 7.

The channel estimation in OFDM systems is addressed in Chapter 5. Again, DA and NDA channel estimators are investigated and compared. In this case, DA methods are based on the inclusion of pilot tones into the OFDM symbols. Three methods, namely Ideal Interpolator, MMSE and Simplified Low-Rank are compared. It is found that the Low-Rank method outperforms the other two if a small number of sub-carriers is used in the OFDM symbols. Nevertheless, the inclusion of an excessive number of pilot tones reduces the spectral efficiency of the transmission scheme. Hence, an NDA approach is further investigated, which is based on a decision-directed channel estimator referred to as CD3 (Coded Decision-Directed Demodulation) firstly proposed by Mignone and Morello in [Mig96]. Since theoretically no pilots are required, this solution is more efficient than the previous ones. Nevertheless, decision-directed systems contain feedback loops and may become unstable under certain circumstances. Methods to reduce the induced loop noise through filters in the frequency and time directions are therefore proposed.

Part II of this Dissertation comprises Chapters 6, 7, 8 and Appendices A and B, and is committed to the implementation issues addressed in the last two goals listed in §1.2. The implementation of a DA synchronizer using the preamble structure as given in the IEEE 802.11a standard is thoroughly described in Chapter 6. Since latency is a major concern in WLAN systems, our novel synchronizer architecture has been optimized to achieve frame detection, fine timing estimation and frequency estimation within the preamble interval. Thus, no data buffering is required at all and latency is reduced. The design is seen to be robust against noise, fading and especially against the transient behavior of the Automatic Gain Control (AGC), which is of special impact in burst transmissions.

Chapter 7 deals with the design of a modified CD3 channel estimator. The CD3 solution studied in Chapter 5 [Mig96] is seen not to be enough for reliable demodulation and decoding of data. In a strict sense, the CD3 channel estimator does not require any pilot tones in a situation where perfect synchronization is achieved. However, this is not the actual case and a Residual Phase Error (RPE) will appear in the OFDM symbols after calculation of the FFT. The four pilot tones contained in the OFDM symbols, according to the 802.11a standard, are intended to estimate the RPE. A brute-force approximation to the problem would suggest the following three steps: 1) Calculation of the phases in the pilot tones through an arctangent operation; 2) linear estimation of the phases affecting the data-carrying tones and; 3) correction through an NCO. Unfortunately, this solution introduces a big latency into the whole digital processing and hence an alternative solution should be enlightened. In our modified CD3 scheme, a novel and highly simplified RPE estimation and correction procedure is presented which is seen to perform as well as the brute-force approximation does. Nevertheless, this procedure is an “a posteriori” method in the sense that it does not correct for the sources of the RPE, but only for the RPE itself. One of these sources is the sampling clock frequency offset, which generates a variable timing offset. At this point, the tracking algorithm proposed in Chapter 4 to track variable timing errors is extended. Hence, the error signal generated by the RPE is further used to generate a new error signal, which is employed to control a variable interpolator [Erup93], [Meyr97]. The main advantage of this approach is that the sampling frequency is not corrected by a digital-analog PLL circuit, but it is entirely corrected in the digital domain. This method simplifies the entire design process, since no complex interactions occurring between the analog and the digital domains have to be taken into account. The variable interpolator is placed immediately after Analog-to-Digital Conversion and is as simple as a 1st-order Farrow structure, since higher order filters have resulted in unstable timing loops.

Chapter 8 is committed to the VLSI implementation of our experimental Baseband Processor. All synthesis results reported there are based on IHP’s 0.25 µm SiGe:C BiCMOS standard cell library (5 metal layers). Initially, the VLSI architecture of the synchronizer is analyzed. In order to reduce the power consumption of the whole design, the synchronizer is divided into two independent data paths, namely tracking data path and processing data path. This enables the separation of the whole design into different clock domains, where each domain is activated when required and remains inactive otherwise. The core layout area of the synchronizer is 13 mm^2, with an estimated power consumption of 140 mW when operated at 20 MHz. The overall latency is 3.9 µs, which is less than one OFDM symbol period (4 µs).

The previous synchronizer design has to be embedded into the Baseband Processor. The design of this Processor was carried out in three different steps in which three different Integrated Circuits (IC), namely BBP1, BBP2 and BBP3, where generated. The BBP1 chip was initially developed to contain only the datapath as defined by the IEEE 802.11a standard. Thus, neither the synchronizer nor the channel estimator where included. Apart from the datapath, the BBP1 includes an Enhanced Parallel Port (EPP) to provide an interface to the upper layers. This initial design was far away from being optimized und resulted in a silicon area of about 107 mm^2 (including pads), with 3.44 Million transistors and an estimated power consumption of 1 W.

The development of the BBP2 chip aims to include not only the synchronizer and the channel estimator but also the decimation and interpolation filters into the BBP1 chip. The elimination of unnecessary buffering in the BBP1 together with a better floor-planning resulted in an actual size of 60 mm^2 (including pads) for the BBP2 chip, with an estimated power consumption of 860 mW in the receive direction and 795 mW in the transmit direction according to the pessimistic estimations given by the Synopsys' Digital Compiler tool. The transistor count for the BBP2 chip is 1.55 Millions.

Finally, in the BBP3 chip both the interpolation and decimation filters have been discarded in order to simplify the testability of the chip. Furthermore, the EPP interface is eliminated since it represents an artificial component, which is not going to be included in our targeted single chip implementation. Thus, the actual size of the BBP3 results in 33.6 mm^2 (including pads). If we consider that the interpolation and decimation filters, together with the EPP interface would add about 10 mm^2 to the actual size of the BBP3 chip, an area reduction of approximately 25% is achieved in this case with respect to the BBP2 chip. The power figures obtained for the BBP3 are as follows: 104 mW for the Tx core, 146 mW for the Rx core and 144 mW for the CLK tree and pads. These figures were obtained making use of the Synopsys' PrimePower^{TM} tool, which calculates the power consumption based on the switching activity inside the chip under certain operating conditions and is more realistic than the previous obtained power figures. These values are very competitive compared to results reported by other companies, which are also very active in this field like Atheros or Toshiba.

Nevertheless, two key blocks have not been included in the previous ICs namely, the variable interpolator for timing correction and the noise-reduction filter inside the CD3 channel estimator loop. This imposes a bound in both the higher modulation scheme and the maximum frame length (in symbols) being supported by our implementation, but it suffices to perform the first functional tests.

The contents in Appendices A and B have been treated separately since they do not follow the natural flow of the Dissertation. Nevertheless, they thoroughly discuss the implementation of two key components in the baseband processing, namely the circular CORDIC processor and the Interpolation/Decimation filters. As explained in Chapter 6, two important operations performed during synchronization, i.e. NCO and Arctangent, are based on the well-known circular CORDIC algorithm, operating in rotational and vectoring mode, respectively. Due to the importance of these operations into the final design, Appendix A is entirely devoted to the implementation of two novel 16-bit pipeline-based circular CORDIC processors.

Initially, a solution for the rotational CORDIC is derived. Two important properties are in the basis of this novel solution. On one hand, it requires no final scale factor compensation, as it is required by the normal

CORDIC implementation. On the other side, the proposed modified rotational CORDIC is able to adaptively select the required basic rotations. The resulting architecture for the rotational CORDIC requires slightly less hardware than the conventional one, but on average it saves 50% computation. After synthesis, the processor core results in an area of 0.7 mm^2, with an estimated power dissipation of 7 mW running at 20 MHz. The latency of the processor is reduced to 14 clock cycles.

The presented scheme is also valid to be used in the vectoring mode of operation, i.e. for calculation of arctangent and absolute magnitude of the input vector, although in this case no adaptive selection of basic rotations is possible. The vectoring CORDIC processor results in an area of 0.5 mm^2 with an estimated power dissipation of 6 mW at 20 MHz.

Finally, Appendix B addresses the problem of designing the digital Interpolation and Decimation filters placed before Digital-to-Analog Conversion and after Analog-to-Digital Conversion, respectively. In our implementation, the OFDM signals are sampled at a rate (80 MHz), which is four times higher than required by the Sampling Theorem. This simplifies the design of the analog Anti-Alias filters, which will require smooth transition bands. Nevertheless, this basically translates the problem from the analog domain to the digital one, where low-pass filters with sharp transition bands are then needed for interpolation and decimation by a factor of four.

Two issues should be considered at this point. On one hand, for the same transition band, lower orders are required for IIR filters than for FIR filters. On the other side, an FIR structure for these filters would be desirable, since they introduce a constant group delay, i.e. a linear phase throughout the different frequencies, thus minimizing the Error Vector Magnitude (EVM) in the constellation diagram. In Appendix B, an algorithm to obtain linear phase IIR filters proposed in [Johan96] is applied in the derivation of our interpolation and decimation filters. Two identical 2nd-order half-band filters are used, each having an attenuation of 20 dB in the stop-band. During reception, these filters are placed immediately after the 1st-order variable interpolator used for timing correction (described in Chapter 7).

Chapter 2

The Orthogonal Frequency Division Multiplex (OFDM) Transmission Scheme

2.1 Introduction

As it was shown in the introductory chapter, the OFDM transmission scheme is gaining a major interest in the design of wireless communication systems. Standards like DAB, DVB, IEEE 802.11a and ETSI BRAN HiperLAN2 make use of OFDM because of its natural robustness to *fading multipath channels*. The present Chapter is committed to establish the working principles of OFDM. Initially, in §2.2 a general model describing the behavior of multipath channels is introduced. The model shows two interesting properties. On one hand, it is suitable for computer simulations; on the other side, it will provide a framework to define some important parameters to be considered when designing OFDM systems. The Chapter is continued in §2.3 with the statement of the mathematical principles supporting OFDM signals. Hence, the definition of orthogonality is revised here for convenience. The Chapter is finalized in §2.4 with a detailed explanation of the PHY layer in the IEEE 802.11a and ETSI BRAN Hiperlan2 standards, since these standards will serve as a reference throughout this Dissertation.

2.2 Channel Model for Wireless Communications

2.2.1 General Description

An important point to consider during the design process of any communications system is the channel being used by this system. In our work, we will restrict ourselves to the wireless transmissions, whether mobile or not. A generalized scenario in a mobile communication has been sketched in Figure 2.1. The transmitted wave may not only follow a direct path towards the receiving antenna, but it will be reflected and attenuated by a number of obstacles (scatters). The signal detected at the receiver's antenna will be therefore a combination of several electromagnetic waves, each having a different phase. The result of the combination will depend on their relative phases, i.e. the difference between the distances traveled by the various waves. This *difference of distances* is measured relative to the wavelength, and so the results obtained for a certain frequency will not be the same for any other. This combination of waves could range from being fully *constructive* to being completely *destructive*. In addition, as the transmitter and receiver are moving at a relative speed from each other, the paths also change with time and a very good reception could turn into a signal breakdown due to a sudden extreme attenuation and vice versa. For this reason wireless channels are referred to as *fading channels*. Such a behavior can be approximately modeled by the following timing response of the multipath channel [Pro95]:

$$h(t,\tau)=\sum_{i=1}^{P(t)} a_i(t)\cdot e^{-j2\pi\cdot f_C\cdot\tau_i(t)}\cdot\delta(\tau-\tau_i(t)), \tag{2.1}$$

where $h(t,\tau)$ is the Channel Impulse Response (CIR); a_i stands for the attenuation and τ_i for the delay in the i-th path. Note that both parameters (a_i, τ_i) are stochastic processes. They may change not only because of a relative speed between the transmitting and the receiving antenna, but also due to sudden appearance of buildings and vehicles or even due to ambient factors like temperature, rain,… In addition, (2.1) considers the CIR being compound of a finite but variable number of scatters ($P(t)$). If the channel has an "infinity of scatters" (diffused multipath), the CIR may then be expressed as

$$h(t,\tau)=a(t,\tau)\cdot e^{-j2\pi f_C\tau}. \tag{2.2}$$

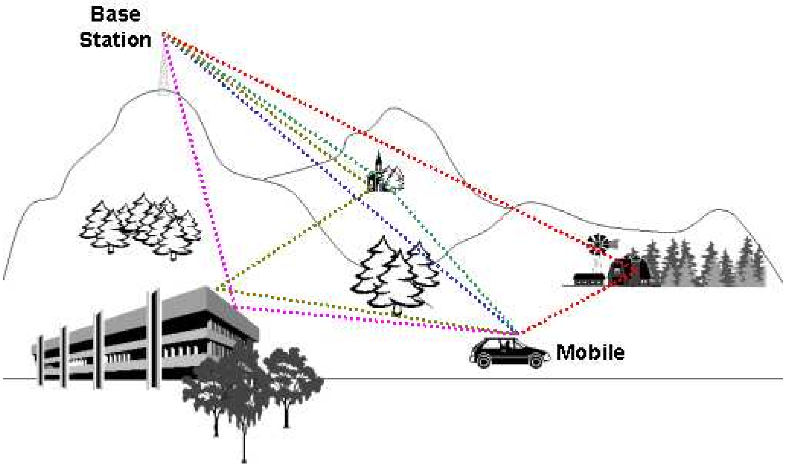


Figure 2.1. Possible scenario in a outdoor mobile communication.

Every realization $a_i(t)$ represents a complex random variable. Each path or ray is considered to be composed of a large number of unresolvable components. Hence, considering the well-known Central Limit Theorem (CLT), $a_i(t)$ can be modeled as a complex Gaussian random variable. In the case of having a large number of paths, the process $h(t,\tau)$ itself will be also Gaussian in the t variable. Thus, $|h(t,\tau)|$ (envelope of $h(t,\tau)$) will be a Rayleigh process at any instant t, as long as no predominant scatter is present. In case of having a line of sight (LOS), i.e. a direct path between the transmitting and receiving antenna, $|h(t,\tau)|$ will be a Ricean process for any t.

2.2.2 Parameter Extraction in Fading Channels

The analysis presented in §2.3 for OFDM signals requires a definition of some important parameters describing a multipath channel [Pro95]. The stochastic process $h(t,\tau)$ is assumed to be wide sense stationary (WSS). This will be the case if the joint probability functions for $h(t_1,\tau)$, $h(t_2,\tau)$, … , $h(t_k,\tau)$ are the same as those for $h(t_1+\Delta t, \tau)$, $h(t_2+\Delta t, \tau)$, … , $h(t_k+\Delta t, \tau)$ for any Δt, for any sequence of times t_1, t_2, … , t_k and for any value of k. Although this is a very restricting requirement, for practical purposes it is enough if the stationary condition is preserved in a certain observation window. Hence, the following autocorrelation function may be calculated,

$$\phi_{\mathrm{h}}(\Delta t;\tau_1,\tau_2) = \frac{1}{2}E\{h^*(t;\tau_1)h(t+\Delta t;\tau_2)\}, \tag{2.3}$$

where the factor ½ comes from the fact that only positive values of τ are considered.

The expression (2.3) may be further simplified by the fact that normally the attenuation and phase shift of the channel associated to the path delay τ_1 are independent of those of the path τ_2 (so-called *uncorrelated scattering*). Under uncorrelated scattering (2.3) may be simplified as follows

$$\frac{1}{2}E\{h^*(t;\tau_1)h(t+\Delta t;\tau_2)\} = \phi_{\mathrm{h}}(\Delta t;\tau_1)\cdot\delta(\tau_1-\tau_2). \tag{2.4}$$

By letting $\Delta t = 0$ in (2.4), the function $\phi_{\mathrm{h}}(\tau)$ is obtained, which is the average output power of the channel as a function of the time delay. This function is normally referred to as the *power delay profile* (PDP) of the multipath channel. In practice, the function $\phi_{\mathrm{h}}(\Delta t;\ \tau)$ is measured by transmitting a wideband RF signal modulated by a pseudo-random sequence and cross-correlating the received sequence with delayed versions of itself. Typically, two parameters are extracted from $\phi_{\mathrm{h}}(\tau)$: the root-mean-squared delay spread τ_{rms}, and the maximum excess delay T_{m}. For the case of a typical exponentially decaying PDP (see Figure 2.2a), τ_{rms} is the time constant of the exponential function, whereas T_{m} is typically defined to be the value of τ for which the exponential is less than 1% of its maximum value. In a general case τ_{rms} is defined as shown in Figure 2.2b.

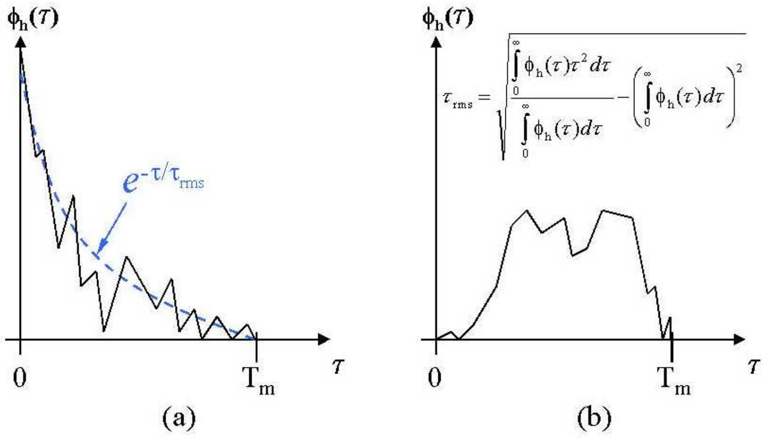


Figure 2.2. (a) Exponential PDP; (b) General PDP with τ_{rms} definition.

Note that T_{m} is giving some indication about the channel bandwidth. Hence, in order to obtain spectral information about the channel, we could calculate a new statistical process by performing the Fourier

Transform of $h(t,\tau)$ with respect to the parameter τ, i.e. $H(t,f) = \mathcal{F}_\tau\{h(t,\tau)\}$. The autocorrelation function for the process $H(t,f)$, i.e. $\phi_H(\Delta t; f_1, f_2)$, can now be obtained in the same way as it was defined in (2.3). It can be seen [Pro95] that under the assumption of uncorrelated scattering, $\phi_H(\Delta t; f_1, f_2)$ does not depend on the exact values of f_1 and f_2, but on its difference $\Delta f = f_2 - f_1$. Furthermore, $\phi_H(\Delta t;\Delta f)$ can be calculated directly by performing the Fourier Transform of $\phi_h(\Delta t;\tau)$ with respect to τ.

Considering the previous statement, it is also possible to find a reciprocal parameter for T_m from the autocorrelation $\phi_H(0;\Delta f) = \phi_H(\Delta f)$. This is the so-called *coherence bandwidth* (B_c) and a typical definition for it would be the bandwidth for which $|\phi_H(\Delta f)|$ is 3 dB below its maximum value. In general it is enough to consider $B_c \approx 1/T_m$. Hence, if the transmitted signal has a bandwidth greater than the coherence bandwidth, it will be severely affected by the channel. In this case the channel is said to be *frequency-selective*. In the opposite case, the channel effects on the signal will be nearly constant in frequency, leading to a *frequency-nonselective* behavior.

Time variations in the wireless channel are mainly caused by the existing relative velocity between transmit and receive antennae. It is well known that in this situation a Doppler "shift" is to be expected. The way this Doppler shift is going to affect our signal can be analyzed by performing the Fourier Transform of the autocorrelation $\phi_H(\Delta t;\Delta f)$ with respect to Δt, i.e. $S_H(\lambda;\Delta f) = \mathcal{F}_{\Delta t}\{\phi_H(\Delta t;\Delta f)\}$. By setting $\Delta f = 0$, the function $S_H(\lambda)$ is a power spectrum that gives the signal intensity as a function of the Doppler frequency λ, thus referred to as the *Doppler power spectrum*. The range of values of λ over which $S_H(\lambda)$ is essentially non-zero is called the Doppler spread B_d. The Doppler spread is directly related to the relative speed between transmitter and receiver by the expression

$$\mathrm{B_d} = v\frac{f_0}{c}, \tag{2.5}$$

where v is the relative speed between transmit and receive antennae, f_0 is the carrier frequency and c is the propagation speed of the electromagnetic wave. The inverse of B_d is called the *coherence time* T_d, and is the mean time during which the channel does not change significantly. If the transmit symbol timing is much shorter than the coherence time, an equalizer can be used at the receiver to compensate for the effects of the time-varying channel; if not, the equalizer will not be able to track the channel variations. In the former situation the channel is said to be *slow fading*, whereas in the later the channel is said to be *fast fading*.

The most commonly used Doppler power spectrum is the Jake's spectrum [Jakes00], [Jeru92], given by the expression

$$S_H(\lambda) = \begin{cases} \dfrac{K}{\sqrt{1-\left(\dfrac{\lambda}{\mathrm{B_d}}\right)^2}}; & -\mathrm{B_d} \le \lambda \le \mathrm{B_d} \\ 0; & \text{otherwise} \end{cases}. \tag{2.6}$$

This Doppler spectrum is valid under the assumptions that all paths have the same Doppler spectrum and all of them reach the receive antenna in a uniform angular distribution. In case of having an LOS situation, the spectrum also includes the effect of the direct path and is termed as the Rician spectrum,

$$S_H(\lambda)=\begin{cases}\dfrac{K_1}{\sqrt{1-\left(\dfrac{\lambda}{\mathrm{B_d}}\right)^2}}+K_2\cdot\left(\delta(\lambda-\cos(\gamma)\mathrm{B_d})+\delta(\lambda+\cos(\gamma)\mathrm{B_d})\right);\ -\mathrm{B_d}\le\lambda\le\mathrm{B_d}\\ 0;\quad \text{otherwise}\end{cases},\tag{2.7}$$

where γ is the angle between the velocity vector of the transmitter and the line connecting the transmitter and the receiver (see Figure 2.3a). Although this angle also changes with time, it could be considered constant when the transmitter and the receiver are separated by a long distance. Figure 2.3b shows the Rician Doppler power spectrum from (2.7), formed by the Jake's spectrum plus two delta functions (LOS components).

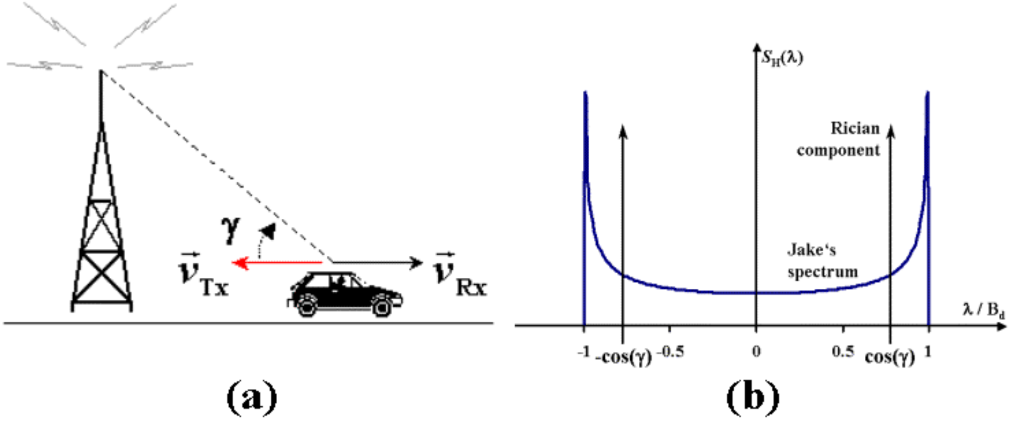


Figure 2.3. (a) Definition of cos(γ) in (2.7); (b) Doppler power spectrum with and without LOS.

Nevertheless, experimental results show that the Jake's Doppler spectrum model does not fit well in indoor scenarios [Thoen02]. The authors in [Thoen02] show that for indoor scenarios the mobile speed does not play a key role, since it is going to be very small. Instead, other phenomena e.g. the movement of the reflecting surfaces make the Doppler spectrum to look rather different to the Jake's spectrum. In this new model, the Doppler bandwidth may extend up to $4\mathrm{B_d}$ and the Doppler spectrum is "peakier" compared to the Jake's one. Nevertheless, since several models have been developed considering a number of possible scenarios without reaching a consensus, we will further consider the Jake's model as our reference model. In Figure 2.4 the different channel correlation functions and their relationship have been summarized.

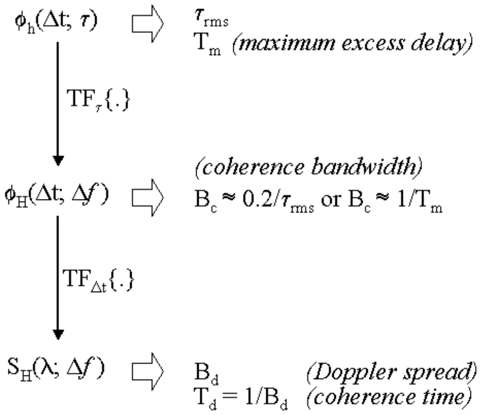


Figure 2.4. Resume of the most important parameters defining a multipath fading channel.

As mentioned above, a small variation in the position of the receiving antenna may lead to a big phase variation in the power of the received signal. It's enough to move half a wavelength to provoke a phase variation of 180°. For that reason mobile systems generally make use of *spatial diversity*. The spatial diversity consists on the use of several antennae placed at suitable distances. The distance between the antennae should be large enough to prevent a big correlation between the wave fronts reaching each antenna. Generally, a separation of at least half the wavelength is enough to assure uncorrelation. During digital processing it is the receiver who decides which antenna will be used for reception (diversity by selection), or even make a phase-corrected and weighted combination of the signals coming from all the antennae (Maximum Ratio Combiner).

It is worthy to mention that such a multipath channel model is not only applicable for wireless communications, but also for systems making use of the power line for digital communications. The unmatched impedances at the different sockets of the electrical network in a home or office environment make the power line to be a communications channel with multipath propagation. Furthermore, the different devices connected to the power line change its impedance with time, depending on whether they are being used or not, or even if they are not plugged at all, making the channel slowly varying.

2.2.3 Simulation Model

Once the most relevant parameters of a wireless mobile channel have been introduced, it is desirable to find a model valid for computer simulations. The model we have made use of is the one explained in [Pro95] and [Jeru92], and is valid for both, diffused or discrete multipath –expressions (2.2) and (2.1), respectively–.

If $x(\mathrm{t})$ is the transmitted baseband signal, the channel output may be expressed as the following convolution

$$y(t,\tau)=\int_{-\infty}^{\infty}h_{lp}(t,\alpha)x(\tau-\alpha)d\alpha\,, \tag{2.8}$$

where $h_{\mathrm{lp}}(t,\tau)$ is the low-pass equivalent CIR. The expression of $h_{\mathrm{lp}}(t,\tau)$ is given by either

$$h_{lp}(t,\tau)=\sum_{i=1}^{P(t)}a_i(t)\cdot\delta(\tau-\tau_i(t))\,, \tag{2.9}$$

or

$$h_{lp}(t,\tau)=a(t,\tau)\,, \tag{2.10}$$

depending on whether a discrete or diffused multipath model is used, respectively.

As $x(\mathrm{t})$ is going to be a band-limited signal with bandwidth B_x, it could be sampled at a rate $2\cdot\mathrm{B}_x$ (time between samples $\mathrm{T}=1/(2\cdot\mathrm{B}_x)$) and expressed as follows

$$x(\tau-\alpha)=\sum_{n=-\infty}^{\infty}x(n\mathrm{T})\frac{\sin\left(2\pi\mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}{\left(2\pi\mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}\,, \tag{2.11}$$

where an ideal low-pass reconstructive filter with bandwidth B_x has been considered. Introducing (2.11) into (2.8) yields

$$y(t,\tau)=\int_{-\infty}^{\infty}h_{lp}(t,\alpha)\left\{\sum_{n=-\infty}^{\infty}x(n\mathrm{T})\frac{\sin\left(2\pi\mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}{\left(2\pi\mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}\right\}d\alpha$$

$$= \sum_{n=-\infty}^{\infty} x(n\mathrm{T}) \left(\int_{-\infty}^{\infty} h_{lp}(t,\alpha) \left\{ \frac{\sin\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}{\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)} \right\} d\alpha \right)$$

$$= \sum_{n=-\infty}^{\infty} x(n\mathrm{T}) g_n(t,\tau)\,, \tag{2.12}$$

with

$$g_n(t,\tau) = \int_{-\infty}^{\infty} h_{lp}(t,\alpha) \left\{ \frac{\sin\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}{\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)} \right\} d\alpha$$

$$\approx \mathrm{T} \cdot h_{lp}(t, n\mathrm{T}) = g(t, n\mathrm{T})\,. \tag{2.13}$$

The approximation done for $g_n(t,\tau)$ in (2.13) will be valid if the sampling interval T << T_m (maximum excess delay of the channel). Furthermore, the range of n can be restricted to the interval $n \in [-m, m]$, yielding

$$y(t, n\mathrm{T}) = \mathrm{T} \sum_{n=-m}^{m} x(n\mathrm{T}) \cdot h_{lp}(t, n\mathrm{T}) = \sum_{n=-m}^{m} x(n\mathrm{T}) \cdot g(t, n\mathrm{T})\,, \tag{2.14}$$

which is an FIR structure with $2m+1$ taps, uncorrelated with each other due to the simplification done in (2.13). The resulting structure is sketched in Figure 2.5.

The analysis done above is strictly valid if the CIR is like in (2.10), i.e. in the case of a diffused scattering. Now we will see that for the discrete approach in (2.9) this is also true. The function $g_n(t,\tau)$ is recomputed by introducing (2.9) into (2.13), yielding

$$g_n(t,\tau) = \int_{-\infty}^{\infty} \sum_{i=1}^{P(t)} a_i(t) \delta(\tau-\alpha-\tau_i(t)) \left\{ \frac{\sin\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}{\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)} \right\} d\alpha$$

$$= \sum_{i=1}^{P(t)} a_i(t) \int_{-\infty}^{\infty} \delta(\tau-\alpha-\tau_i(t)) \left\{ \frac{\sin\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)}{\left(2\pi \mathrm{B}_x(\tau-\alpha-n\mathrm{T})\right)} \right\} d\alpha$$

$$= \sum_{i=1}^{P(t)} a_i(t) \frac{\sin\left(2\pi \mathrm{B}_x(\tau_i(t)-n\mathrm{T})\right)}{\left(2\pi \mathrm{B}_x(\tau_i(t)-n\mathrm{T})\right)}$$

$$= \sum_{i=1}^{P(t)} a_i(t) \frac{\sin\left(\pi(\frac{\tau_i(t)}{\mathrm{T}} - n) \right)}{\left(\pi(\frac{\tau_i(t)}{\mathrm{T}} - n) \right)} = \sum_{i=1}^{P(t)} a_i(t) \alpha(t,i,n) = g(t, n\mathrm{T})\,, \tag{2.15}$$

where the function $\alpha(t, i, n)$ includes the fact that the defined delays for each path may not be multiples of the selected sampling time T.

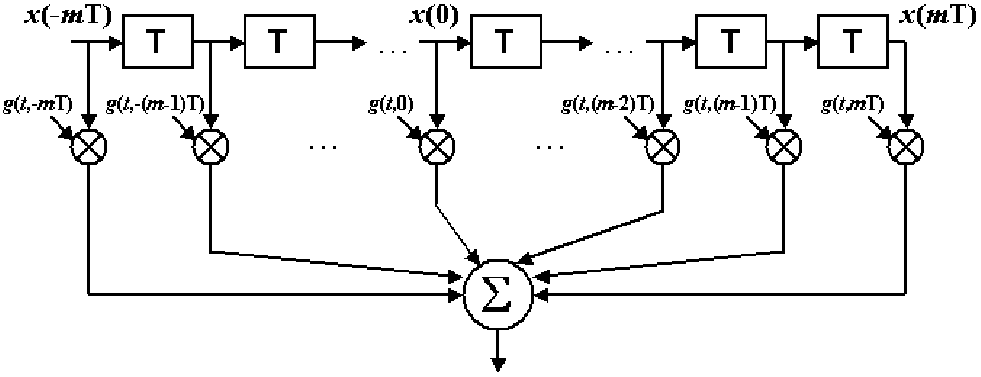


Figure 2.5. FIR structure for the channel simulator.

The coefficients $g(t, n\mathrm{T})$ in (2.15) are complex Gaussian random variables with a power spectrum density given by (2.6). The way to generate these coefficients is shown in Figure 2.6. For each coefficient a random number generator is required to produce white Gaussian noise with zero mean and unit variance. The noise is filtered by a Doppler-filter with a transfer function given by

$$H_d(f)=\begin{cases}\sqrt{\dfrac{K}{\sqrt{1-\left(\dfrac{f}{\mathrm{B_d}}\right)^2}}}; & -\mathrm{B_d}\le f\le \mathrm{B_d}\\ 0; & \text{otherwise}\end{cases}, \tag{2.16}$$

or equivalently, an impulse response as

$$h_d(t)=K\cdot \mathrm{B_d}\cdot\sqrt[4]{\frac{2}{2\pi \mathrm{B_d} t}}\cdot\Gamma(3/4)\cdot J_{1/4}(2\pi \mathrm{B_d} t), \tag{2.17}$$

where $J_{1/4}$ is the fractional Bessel function and Γ is the gamma function. The impulse response h_d(t) has been represented in Figure 2.7. The energy of the filter in (2.16) is obtained after integration in the interval $[-\mathrm{B_d}, \mathrm{B_d}]$, resulting in $\mathrm{E_d} = K\cdot\pi$.

In a real application, the bandwidth of the Doppler filter is very small ($\mathrm{B_d}$ = 1,740 Hz for f_C = 5 GHz and v=100 km/h) compared to the bandwidth of the input signal, thus having two rather different sampling rates inside the filter in Figure 2.5. A way to compensate this difference is by interpolating the coefficients g(t, nT), as shown in Figure 2.6. However, the interpolation factor will be very big, meaning that large filter orders are required to obtain worthy results. Polyphase filters are commonly used to implement this interpolation in a more efficient way [SPW]. Furthermore, the simulation model depicted in Figure 2.5 is valid for channels without LOS. If a rice component is present, the tap $g(t,0)$ in Figure 2.5 is generally taken as

$$g(t,0)=e^{j2\pi\cdot\cos(\gamma)\cdot \mathrm{B_d}\cdot t}, \tag{2.18}$$

where γ is given as in Figure 2.3a.

The reference channel models proposed by the ETSI in [ETSIb] for the 5 GHz band (see Appendix D) match very well the channel simulation model presented so far. Therefore, this model will be systematically used in this Dissertation to analyze the performance of the proposed architectures.

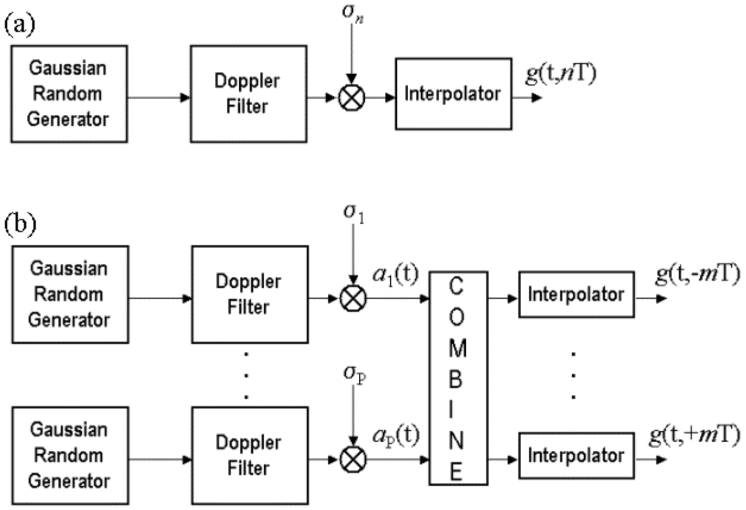


Figure 2.6. Generation of filter coefficients for (a) diffused multipath; (b) discrete multipath.

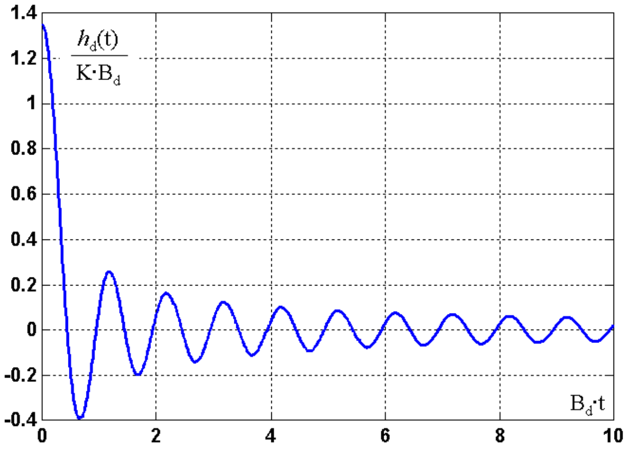


Figure 2.7. Impulse response of the Jakes' spectrum modeling filter.

2.3 OFDM Fundamentals

2.3.1 Short History of OFDM

The main idea behind OFDM is the so-called Multi-Carrier Modulation (MCM) transmission technique. MCM is the principle of transmitting data by dividing the input bit stream into several parallel bit streams, each of them having a much lower bit rate, and by using these sub-streams to modulate several carriers as depicted in Figure 2.8. The first systems using MCM were military high frequency radio links in the late 1950s and early 1960s, like Kineplex, Andeft and Kathryn. OFDM is a special form of MCM with densely spaced sub-carriers and overlapping spectra, whose main idea was patented by Chang, from the Bell Labs, in 1966. OFDM abandoned the use of steep bandpass filters that completely separated the spectrum of individual sub-carriers, as it was common practice in older analogue FDMA systems. Instead, OFDM time-domain waveforms are chosen such that mutual *orthogonality* is ensured even though carrier spectra may overlap. Orthogonality is achieved by performing a Fourier Transform (or equivalently a Fast Fourier Transform) on the input stream.

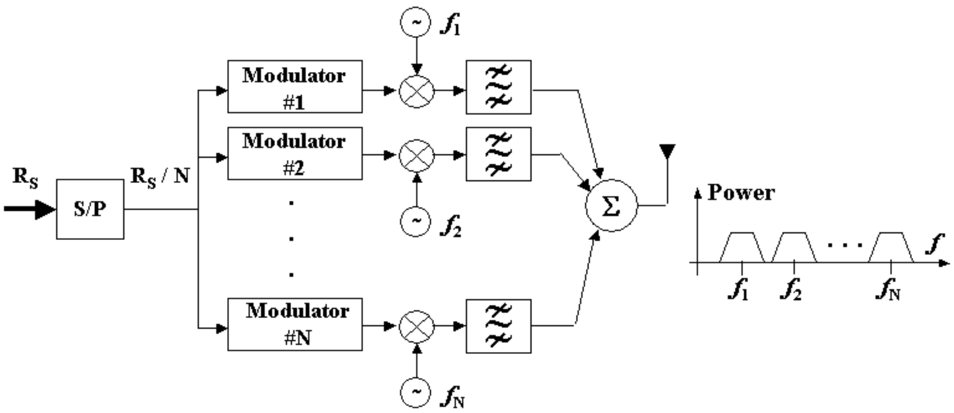


Figure 2.8. Block diagram of a generalized MCM transmitter.

Implementation aspects have delayed the introduction of OFDM into real systems. For instance, the complexity of a real-time FFT requires powerful digital signal processors, which were not available at the time when OFDM was invented. Furthermore, in order to keep the orthogonality between the different sub-carriers, highly stable oscillators are required in the transmitter and receiver, as well as linear power amplifiers. Nowadays many of the implementation problems appear solvable and OFDM has gained a big interest since the beginning of the 1990s [Bing90]. This interest is mainly motivated by the advantages OFDM offers when transmitting through a fading channel. One parameter characterizing these channels is the *maximum delay spread* (T_m), as defined in §2.2.2. By transmitting N parallel bit streams, the symbol duration is increased and so the relative delay spread decreases. Table 2.1 provides a short chronology of the most important contributions in OFDM during the last years.

Table 2.1. Short chronology of OFDM.

Year	*Event*
1957	Kineplex multi-carrier HF modem
1966	Chang (Bell Labs): OFDM paper and patent
1971	Weinstein & Ebert propose the use of FFT with guard interval
1985	Cimini describes use of OFDM for mobile communications
1987	Alard & Lasalle: OFDM for digital broadcasting
1995	ETSI DAB standard: first OFDM-based standard
1997	DVB-T standard
1998	Magic WAND project demonstrates OFDM modems for wireless LAN
1999	IEEE 802.11a and ETSI BRAN Hiperlan/2 standards for wireless LAN
2000	V-OFDM for Fixed Wireless Access
2001	OFDM considered for new 802.11 and 802.16 standards
2003	OFDM considered for UWB (802.15.3a)

2.3.2 Mathematical Model of the OFDM Signal

In order to perform the mathematical analysis of OFDM signals, the most convenient strategy is to use the signal theory related to the orthogonal expansion of signals [Pro95]. In this theory, the signal is contained in a subspace and is represented by a linear combination of the *base signals* $\Psi_n(t)$ of that subspace. The signals conforming the base have the property of being orthonormal, i.e.

$$\int_{-\infty}^{\infty} \Psi_n(t)\Psi_m^*(t)dt = \begin{cases} 0\,(m \neq n) \\ 1\,(m = n) \end{cases}, \tag{2.19}$$

and any signal $s(t)$ may be represented in the following lineal form

$$s(t) = \sum_{k=0}^{\mathrm{N}-1} A_k \Psi_k(t)\,, \tag{2.20}$$

where the coefficients A_k are complex. As we already mentioned in §2.3.1, OFDM is a special case of MCM (see Figure 2.8), but with the special feature that the different sub-carriers are allowed to overlap their spectra. In the case of OFDM, the system is sketched in Figure 2.9. From Figure 2.9 it is straightforward to identify the kind of base signals being used, i.e.

$$\Psi_k(t) = w(t)e^{j2\pi\left(f_0 + k\cdot\Delta f\right)\cdot t}, \tag{2.21}$$

where $w(t)$ is a window of length T. The easiest choice for $w(t)$ is the normalized rectangular window defined as

$$w(t) = \begin{cases} \dfrac{1}{\sqrt{\mathrm{T}}}\,;\ |t| \leq \mathrm{T}/2 \\ 0\,;\ \text{otherwise} \end{cases}. \tag{2.22}$$

Considering the window in (2.22) and applying the condition given by (2.19), yields

$$\begin{aligned} \varepsilon_\perp &= \int_{-\infty}^{\infty} \Psi_n(t)\Psi_m^*(t)dt = \frac{1}{\mathrm{T}}\int_{-\mathrm{T}/2}^{\mathrm{T}/2} e^{j2\pi\cdot\Delta f(n-m)\cdot t}dt \\ &= \frac{e^{j2\pi\Delta f(n-m)(\mathrm{T}/2)} - e^{-j2\pi\Delta f(n-m)(\mathrm{T}/2)}}{j2\pi\Delta f(n-m)\mathrm{T}} \\ &= \frac{\sin(\pi\Delta f(n-m)\mathrm{T})}{\pi\Delta f(n-m)\mathrm{T}}. \end{aligned} \tag{2.23}$$

For the case $m = n$, the expression (2.23) directly yields $\varepsilon_\perp = 1$. The orthogonality is achieved by setting $\Delta f = 1/\mathrm{T}$, thus making $\varepsilon_\perp = 0$ for any $m \neq n$. The window $w(t)$ in (2.22) was selected to be the rectangular pulse only for simplicity. Considering this window, the Fourier transformation of the signal $s(t)$ in (2.20) will be made up by a combination of N *sinc*[1] functions, each one separated by 1/T (A_k are considered deterministic constants), i.e.

$$S(f) = \sqrt{\mathrm{T}}\sum_{k=0}^{\mathrm{N}-1} A_k \operatorname{sinc}\left(\mathrm{T}(f - f_{\mathrm{o}} - \frac{k}{\mathrm{T}})\right). \tag{2.24}$$

---

[1] sinc(x) ≡ sin(πx)/πx

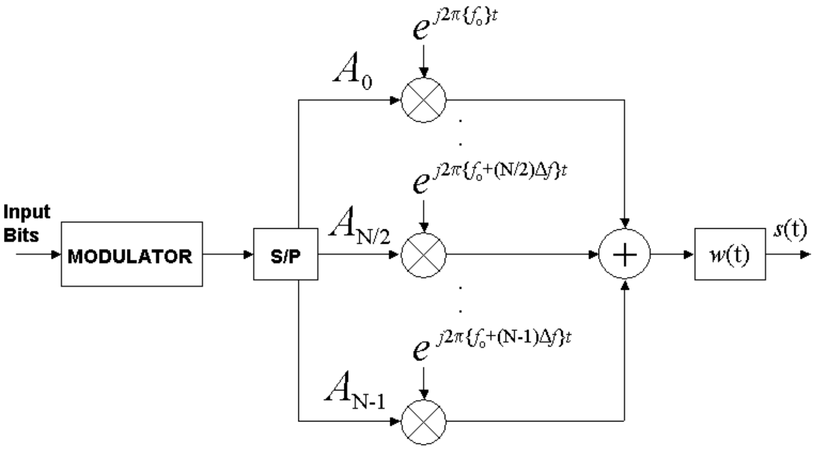


Figure 2.9. General scheme of an OFDM transmitter.

The total bandwidth occupied by the OFDM signal may be obtained from (2.24). This bandwidth directly depends on the selected window *w*(*t*). In this particular case, the *sinc* function decreases as 1/*f*, thus generating big secondary lobes that extend the bandwidth. Hence, the total bandwidth may be reduced by considering other windows with smaller secondary lobes.

Smaller secondary lobes can be achieved by letting the window *w*(*t*) be longer than T, e.g.

$$w(t) = \begin{cases} f(\mathrm{t})\,; -\frac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}}) \le t \le \frac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}}) \\ 0\,; \text{otherwise} \end{cases}. \tag{2.25}$$

The orthogonality condition in (2.19), considering signals as in (2.21) with Δf = 1/T, will be kept if the following expression is satisfied,

$$\int_{-\frac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})}^{\frac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})} |w(t)|^2 e^{j\frac{2\pi}{\mathrm{T}}kt} dt = 0;\ (k \ne 0), \tag{2.26}$$

or equivalently,

$$\int_{0}^{\frac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})} |w(t)|^2 \cos\left(\frac{2\pi}{\mathrm{T}}kt\right) dt = 0;\ (k \ne 0). \tag{2.27}$$

The functions $|w(\mathrm{t})|^2$ which accomplish the condition in (2.27) are those having a *vestigial symmetry* around *t*=T/2. The definition of vestigial symmetry is given graphically in Figure 2.10, where the integral in (2.27) has been represented for the particular case of *k*=1. Nevertheless, due to the even symmetry of the cosine function around *t* = T/2 in (2.27), the integral will be zero for any integer value of *k* ($k \ne 0$). There is an infinite number of functions satisfying the previous condition. Besides the rectangular window in (2.22), we will mention the *linear* pulse (2.28) and the *raised cosine* pulse (2.29). The two later windows are controlled by a parameter α_r called roll-off factor, $0 \le \alpha_r \le 1$, which determines the bandwidth of the window,

$$w_{\mathrm{LIN}}^{2}(t)=\begin{cases}\dfrac{1}{\mathrm{T}}\,;\ |t|\le\dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\\ \dfrac{1}{\mathrm{T}\alpha}\left(-\dfrac{|t|}{\mathrm{T}}+\dfrac{1}{2}(1+\alpha_{\mathrm{r}})\right);\ \dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\le|t|\le\dfrac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})\\ 0\,;\ \text{otherwise}\end{cases},\tag{2.28}$$

$$w_{\mathrm{RC}}^{2}(t)=\begin{cases}\dfrac{1}{\mathrm{T}}\,;|t|\le\dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\\ \dfrac{1}{2\mathrm{T}}\left\{1+\cos\left(\dfrac{\pi}{\alpha_{\mathrm{r}}}\left(\dfrac{|t|}{\mathrm{T}}-\dfrac{(1-\alpha_{\mathrm{r}})}{2}\right)\right)\right\};\ \dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\le|t|\le\dfrac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})\\ 0\,;\ \text{otherwise}\end{cases}.\tag{2.29}$$

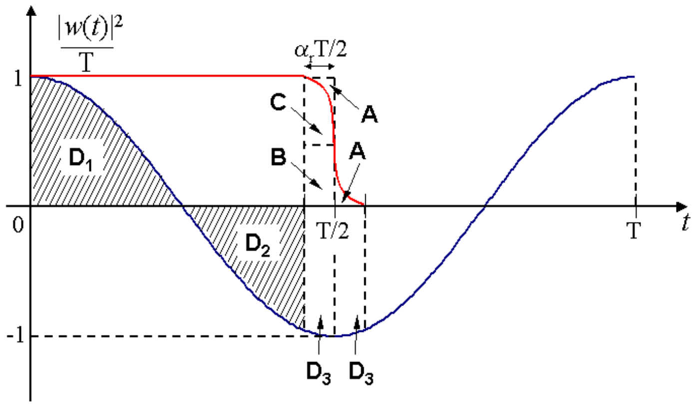


Figure 2.10. Graphical interpretation of the vestigial symmetry.

Note that the previous expressions represent the square value of the respective window. For the special case of $\alpha_r = 0$, both windows (2.28) and (2.29) result in the rectangular one. The representation of these windows is sketched in Figure 2.11, for the particular case of $\alpha_r = 0.4$.

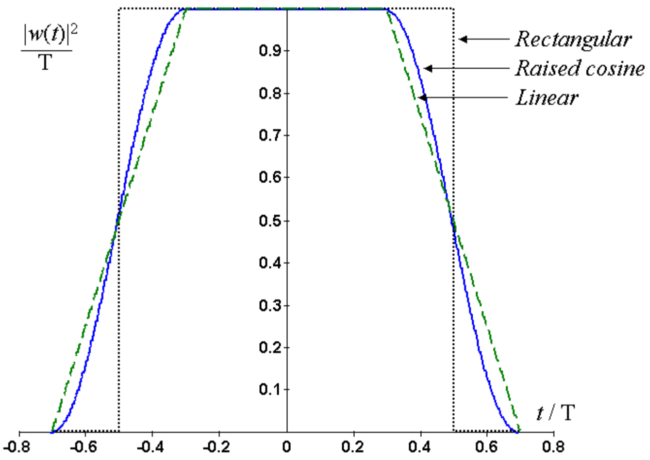


Figure 2.11. Representation of normalized windows: rectangular, raised cosine and linear ($\alpha_r = 0.4$).

The analysis of the spectrum of these windows in Figure 2.12 shows that all of them have zero-crossings at integer values of 1/T, thus achieving orthogonality. In addition, the linear and raised cosine pulses achieve a better spectral efficiency because of their lower secondary lobes.

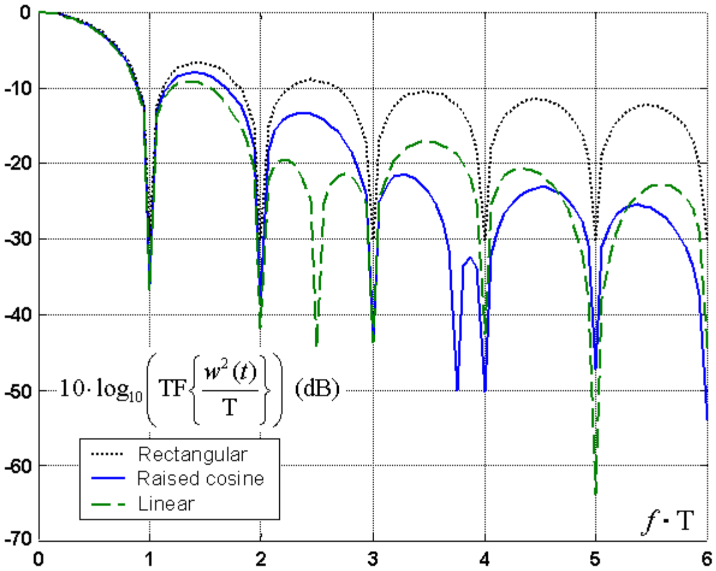


Figure 2.12. Comparison of the spectra for different normalized windows ($\alpha_r = 0.4$).

During transmission of the OFDM signal, the symbols will be multiplied by the window $w(t)$ and not by $w^2(t)$. The spectra of the pulses $w(t)$ have no zero-crossings at integer values of 1/T as it happens with $w^2(t)$ –it only happens for the particular case of the rectangular window, because in this case $w(t) = w^2(t)$–. The orthogonality is achieved at the receiver side by multiplying the incoming signal by $w^*(t)$. The information sequence A_k is obtained after multiplication by the corresponding complex conjugated phasor and integrating over an interval $T(1+\alpha_r)$, as shown in Figure 2.13. In Figure 2.14, the spectrum of an OFDM signal with $A_k=1$ $\forall k$ and N=32 is represented. There it can be better appreciated the dependency of the total bandwidth of the OFDM signal with respect to the selected window. Clearly, the root raised cosine (RRC) window results in the better bandwidth efficiency among the proposed windows.

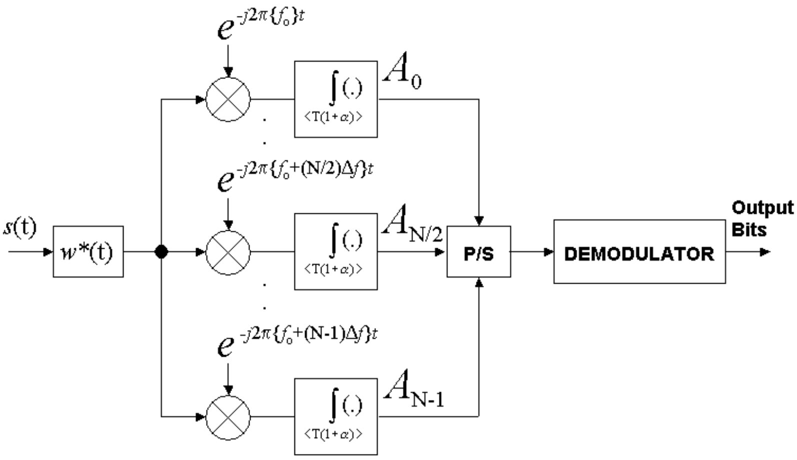


Figure 2.13. General scheme of an OFDM receiver.

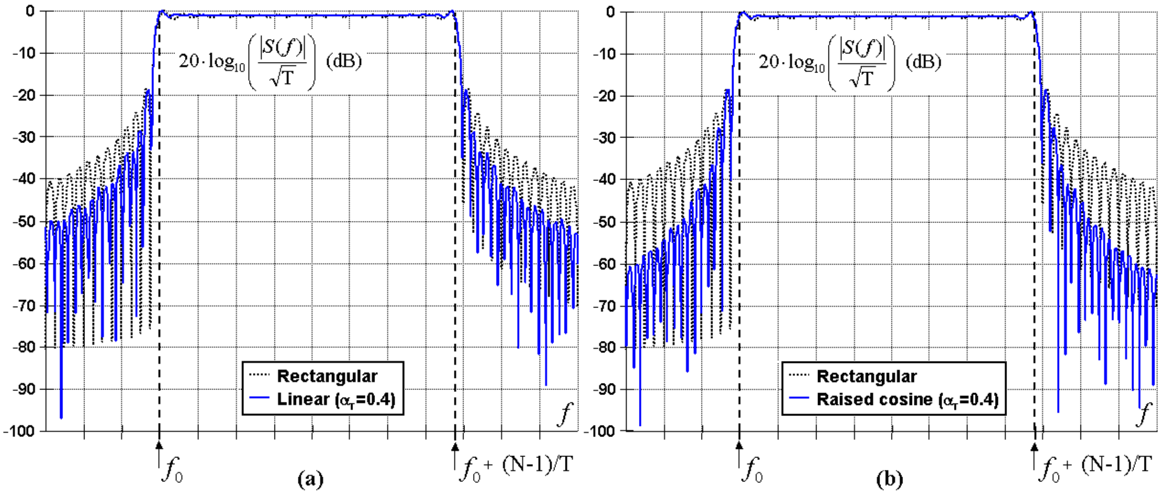


Figure 2.14. Representation of the spectrum of an OFDM signal with N=32, using rectangular, square-root linear and square-root raised cosine windows (α_r = 0.4).

However, there is a big drawback associated with the parameter α_r. As the symbols are being extended, some of the information of one symbol will spread out into the neighboring symbols, thus causing Inter-Symbol Interference (ISI). In the following section, it is clarified how to cope with this problem.

2.3.3 Digital Implementation of an OFDM System

For the practical implementation of a digital OFDM signal generator, it is firstly considered that the window $w(t)$ is rectangular and the parameter $f_0 = -(N-1)/2T$ in (2.21). By this, the spectrum of the transmitted signal is centered at $f = 0$ and spans from $-(N-1)/2T$ ($n = 0$) to $+(N-1)/2T$ ($n = N-1$). The highest frequency is then $(N-1)/2T$. Considering the Sampling Theorem, a sampling frequency being at least twice the highest frequency should be selected. Setting $f_S = N/T$, (2.20) yields

$$s(n\frac{T}{N}) = w(n\frac{T}{N})\sum_{k=0}^{N-1} A_k e^{j\frac{2\pi}{N}kn} = N\cdot e^{-j\frac{2\pi(N-1)n}{N}}\cdot w(n)\cdot \mathrm{IDFT}\{A_k\} = C\cdot w(n)\cdot a_n, \tag{2.30}$$

for $0 \le n \le N-1$.

Thus the whole operation can be performed efficiently using either the FFT algorithm in case of N being a power of two, or the CORDIC algorithm otherwise. This form of generation of OFDM signals is generally referred to in the literature as Discrete Multitone (DMT). Note that the symbols A_k have a rate of 1/T baud, but the OFDM symbols will be transmitted at a rate of 1/(NT) baud.

Equivalently, at the receiver side a DFT operation will be carried out, i.e.

$$A_k = \mathrm{DFT}\{w^*(n)\cdot s(n)\}, \tag{2.31}$$

for $0 \le k, n \le N-1$.

In case of using a window different of the rectangular one, the OFDM symbols will be longer than N samples, but still the sequence A_k will contain only N samples. The way to resolve this situation is by adding a *prefix* and a *suffix* (or *postfix*) of length α_r(N/2) each to the sequence a_n, [Nee00]. The prefix and the suffix are obtained after windowing a *periodic extension* of the IDFT output, defined as

$$\tilde{a}_n = \sum_{r=-\infty}^{\infty} a_{n+rN} \,. \tag{2.32}$$

With this modification, (2.30) yields

$$s(n) = C \cdot w(n) \cdot \tilde{a}_n \,, \tag{2.33}$$

with $0 \le n \le N + \alpha_r N - 1$. On the receiver side, the prefix and suffix samples should be removed prior to the DFT calculation, in order to avoid the ISI.

One of the most important practical reasons to use OFDM is the efficient way it deals with multipath propagation. When an OFDM signal is transmitted through a channel as in (2.1), the length of the symbols is extended by T_m (being T_m the maximum excess delay of the channel), thus introducing ISI from one symbol to the next one. A simple solution to avoid ISI might be to leave a time period T_m between symbols where nothing is transmitted at all. Nevertheless, this is not a good solution in terms of bandwidth efficiency, since high frequency components are generated in the transmitted signal due to the abrupt discontinuities in the transmitted power. In addition, this solution may add important difficulties considering the synchronization problem of the OFDM symbols, as any error in the positioning of the DFT (FFT) window will lead to a loss of orthogonality and therefore Inter-Carrier Interference (ICI) will appear. The inclusion of a cyclic prefix longer than T_m may solve both problems in one shot, as it was firstly shown by Weinstein and Ebert in [Wein71].

As defined in §2.2.2, a channel is said to be frequency selective if its coherence bandwidth $B_c \approx 1/T_m$ is much smaller than the signal bandwidth. In OFDM, one may select the frequency separation of the different sub-carriers to be much smaller than B_c, thus transforming a frequency selective channel into a frequency non-selective channel by setting $1/T << 1/T_m$ or equivalently $T >> T_m$. In the same way, a channel is said to be time-selective if its coherence time T_d is much smaller than the symbol time. In order to be able to equalize the channel, the symbol timing should be much smaller than the coherence time, i.e. $T << T_d$. This fact sets a limitation to the maximum speed allowed by the system when using OFDM signals.

The OFDM transmission scheme can also help to simplify the *channel equalization* necessary during reception. The analog signal being received is the result of the convolution between the transmitted signal $s(t)$ and the channel response $h(t)$, i.e. $r(t) = s(t)*h(t)$. In the frequency domain this is equivalent to multiply their frequency responses $S(f)\cdot H(f)$. For our purpose, it is of interest to know what happens in the frequency domain, since a DFT operation will be carried out during reception. Here we recall two important properties regarding the DFT and the Fourier Transform (FT), [Opp99]:

1. The first property links both transforms and states that given a sequence $x(n)$ with non-zero values <u>only</u> in the range $n \in [0, N-1]$, its corresponding FT ($X(e^{j\omega})$) and DFT ($X(k)$) are related as

$$X(k) = X(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k} \,, \tag{2.34}$$

   i.e. the DFT is obtained after sampling the FT at intervals $2\pi k/N$, $k = 0,\ldots N-1$.
2. The equivalent of the linear convolution in the FT is the circular convolution in the DFT.

(FT) $$x(n) * y(n) \leftrightarrow X(e^{j\omega})\cdot Y(e^{j\omega}) \tag{2.35a}$$

(DFT) $$x(n) \otimes y(n) \leftrightarrow X(k)\cdot Y(k) \tag{2.35b}$$

with

$$x(n) \otimes y(n) = x(n) * \sum_{r=-\infty}^{\infty} y(n - r\mathrm{N}), \tag{2.36}$$

for $0 \le n \le \mathrm{N} - 1$.

The first condition will be satisfied, since each OFDM symbol is multiplied by a conforming pulse $w(t)$ (rectangular pulse for this particular case), which will set to zero all those samples outside the symbol limits. Considering the second condition in (2.35a/b), the circular convolution is defined as the linear convolution of one of the sequences by the periodic extension of the other sequence, with period N. The resulting sequence is also periodic, but only one period is kept, i.e. $n = 0\ldots\mathrm{N} - 1$. This periodic extension may be added artificially at the transmitter side by generating a cyclic prefix, which will force the linear convolution with the channel response to be equivalent to a circular convolution.

As analyzed in Appendix C, a correct cyclic extension helps in keeping the orthogonality inside the symbol, since each sub-carrier at the receiver side can be expressed as $A_{\mathrm{R}k} = H_k \cdot A_k$, where H_k corresponds to the value of the CTF at sub-carrier k. This property simplifies very much the channel equalization on the receiver side in comparison with the traditional single carrier transmissions, where an inverse convolution has to be carried out. The only condition for that is that the cyclic prefix length must be longer than the CIR.

Nevertheless, it should be noticed that the cyclic extension feature will be destroyed if the conforming pulse $w(t)$ is not the rectangular pulse. This is solved by taking a prefix of length $\mathrm{N}((\alpha_r/2)+(\mathrm{T_m}/\mathrm{T}))$ samples, whereas the suffix is kept to be $\alpha_r(\mathrm{N}/2)$ samples, as shown in Figure 2.15.

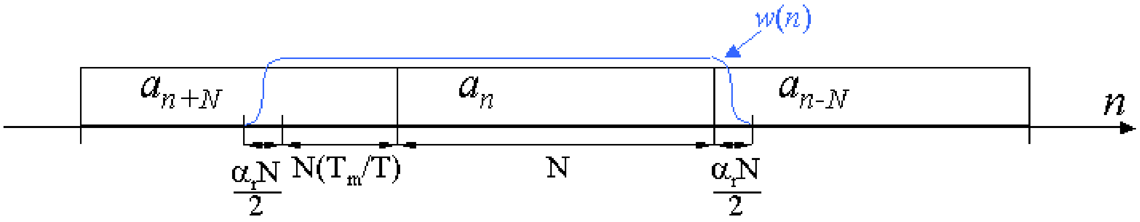


Figure 2.15. Graphic interpretation of the prefix and suffix insertion in an OFDM symbol.

The analysis carried out throughout this section provides us the basic knowledge to specify the different parameters of an OFDM-based system. An example for such a system is provided in the next section, where all the features regarding the PHY layer in the IEEE 802.11a and ETSI HiperLAN2 systems are described.

2.4 Case of Study: the IEEE 802.11a and ETSI HiperLAN2 Standards

2.4.1 General Model of the Physical Layer

The standards IEEE 802.11a [IEEE] and ETSI BRAN HiperLAN2 [ETSIa] are directed to Wireless Local Area Networks (WLAN) working in the 5 GHz ISM band. Both standards make use of an OFDM transmission scheme and have very similar physical layers. Figure 2.16a/b sketches the common model of the baseband transmitter. The parameter *rate* shown there indicates the nominal transmission rate and determines the behavior of the convolutional encoder, the interleaver as well as the mapper. In Table 2.2, the different code rates and modulation schemes are listed for the different possible rates in both standards. A brief description of the several blocks is provided below:

Scrambler: The main purpose of the scrambler is to randomize the input bits, aiming to obtain the same bit probability at its output. A block diagram can be found in Figure 2.17a. The state parameters $x_0,\ldots,x_6$ are initialized using a pseudo-random generator and set to a non-zero value. In the case of the HL2 standard, x_6, x_5 and x_4 are always initialized to '1'.

Convolutional encoder: The convolutional encoder performs the first part of the channel coding. The structure is given in Figure 2.17b. This block introduces some correlation between several consecutive bits in order to increase the robustness of the system towards the noise, considering that the noise will be by nature uncorrelated with the bits. On the receiver side, the decoder will exploit this correlation to extract the correct bits from the noise. The code length is K=7, i.e. the encoder can take 64 possible states, and the code rate is Rc=1/2. Nevertheless, depending on the *rate* parameter, the encoder applies other code rates, i.e. 2/3, 3/4 and 9/16, by using a "bit-stealing" algorithm. This procedure is shown in more detail in Figure 2.18. Note that for each input bit b_x, two coded bits A_x and B_x are obtained after the encoder. The "bit-stealing" algorithm following the encoder will eliminate one of the coded bits but never both of them. At the receiver side, a "bit-insertion" algorithm is applied prior to decoding. Generally, this algorithm inserts alternatively either a '1' or a '0' at those positions where the encoder eliminated the bits. This bit uncertainty is resolved by the decoder, which in this case may be selected to be the Viterbi decoder since K (code length) is not very big.

Interleaver: The second part of the channel coding is used to protect the data against the bursty errors found in the channel. Since data is being transmitted using different sub-carriers, in case the channel degrades one of the sub-carriers very much, it is very likely that neighboring sub-carriers are also being degraded. As previously mentioned, the convolutional encoder introduces some correlation into the bits. If a substantially big number of bits are erroneous, the correlation gets destroyed, and the decoder is no longer able to perform correctly. Hence, the solution is to transmit the coded bits into conveniently separated sub-carriers. The interleaver takes groups of N_{CBPS} (coded bits per OFDM symbol, as shown in Table 2.2) and applies two permutations. The first permutation is given by the following expression:

$$i = (N_{CBPS} / 16) \cdot (k \bmod 16) + floor(k / 16), \qquad (2.37)$$

where k is the index of an incoming bit, $0 \le k \le N_{CBPS}-1$, i is its index at the output and $floor(x)$ keeps the integer part of the real number x. The second permutation is defined by the expression:

$$j = s \cdot floor(i / s) + (i + N_{CBPS} - floor(16 \cdot i / N_{CBPS})) \bmod s, \qquad (2.38)$$

where j is the index after the second permutation. The parameter s is 1 for BPSK and QPSK, 2 for 16-QAM and 3 for 64-QAM.

Mapper: This block maps the input bits using one of the following coherent amplitude modulations: BPSK, QPSK, 16-QAM, 64-QAM. The block takes groups of N_{CBPS} coded and interleaved bits, and generates groups of 48 complex values as shown in Figure 2.19, with the form $K_{MOD}\cdot(I+jQ)$. The parameter K_{MOD} is a normalization factor being 1 for BPSK, $1/\sqrt{2}$ for QPSK, $1/\sqrt{10}$ for 16-QAM and $1/\sqrt{42}$ for 64-QAM.

OFDM transmission block: The information is transmitted into N = 64 sub-carriers, enumerated as $k = -32, \ldots, +31$, i.e. $f_0 = -N/2T$ in (2.21). The frequency separation between sub-carriers is $\Delta f = 1/T = 312.5$ kHz. The baseband signal will extend its bandwidth from $-N/2T$ to $(N/2T)-(1/T)$, i.e. from –10 MHz to +9.6875 MHz. Nevertheless, not all the 64 sub-carriers will carry user information, but only 48 are used as *data* sub-carriers, i.e. the ones corresponding to $k = [-26, -22]\cup[-20, -8]\cup[-6, -1]\cup[+1, +6]\cup[+8, +20]\cup[+22, +26]$. There are 12 further sub-carriers permanently set to zero (*virtual* sub-carriers) at positions $k = [-32, -27]\cup[0]\cup[+27, +31]$. This gives some advantages during implementation: on one hand the DC level (k=0) is set to zero to avoid problems with the RF part; on the other side, it reduces the adjacent channel interference by setting to zero the most peripheral sub-carriers.

The last four remaining sub-carriers corresponding to $k = -21, -7, +7, +21$ are defined as *pilots*. The purpose of the pilots is to simplify some of the operations realized at the receiver, mainly during channel estimation. More details on these operations are seen in Chapters 5 and 7 of this Dissertation, where a generalized study of the channel equalization in OFDM systems is presented. The *pilot generation* shown in Figure 2.16b is realized using the scrambler shown in Figure 2.17a. In this case however, the scrambler is initialized to the “all-ones” state and with a permanent '0' bit at the input. Under these conditions, the scrambler generates a 127-bit periodic pseudo-random sequence. The pilots at $k = -21$, -7 and $+7$ are set to 1 if the scrambler generates a '0', whereas they are set to –1 if the scrambler generates a '1'. The pilot at $k = +21$ is sign-reversed with respect to the others.

IFFT: Since N is a power of 2, an FFT/IFFT algorithm is used to implement the DFT/IDFT. The FFT time is $T = 1/\Delta f = 3.2\ \mu s$ and the sampling frequency is $(N/T) = 20$ MHz. As previously mentioned, the parameter f_0 in (2.21) is $-N/2T$. This value of f_0 can be achieved in two different ways: 1) by setting appropriately the central frequency during RF up-conversion or; 2) by crossing the inputs to the IFFT block, as shown in Figure 2.16b.

Cyclic Prefix (CP) insertion: This block is used to introduce a prefix and a suffix and to window the OFDM symbols. In these standards, the suffix is one sample long, whereas the prefix is 16 samples long. One of these prefix samples is used for the windowing and the remaining 15 are used to compensate for the channel delay spread. The maximum allowed delay spread (T_m) is therefore $(15/64)\cdot 3.2\mu s = 750$ ns, which is equivalent to a path of about 200 meters. The whole OFDM symbol is therefore 81 samples long. The selection of the conforming pulse $w(n)$ is left to the designer. Nevertheless the standards suggest the window $w(n)$ being 0.5 for $n = 0, 80$ and 1 for $1 \le n \le 79$.

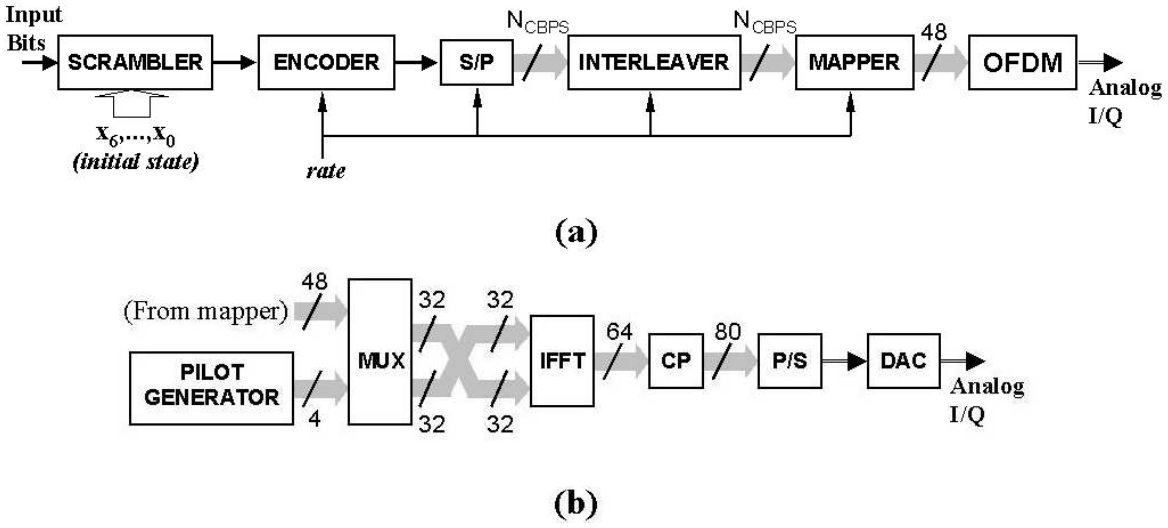


Figure 2.16. Block diagram of the baseband transmitter for the IEEE 802.11a and ETSI BRAN Hiperlan2: a) General scheme; b) Particular OFDM transmission scheme.

Table 2.2. Rate dependent parameters.

Nominal rate (Mbps)	Coding rate (Rc)	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Modulation
6	1/2	48	24	BPSK
9	3/4	48	36	BPSK
12	1/2	96	48	QPSK
18	3/4	96	72	QPSK
24	1/2	192	96	16-QAM
27*	9/16	192	108	16-QAM
36	3/4	192	144	16-QAM
48**	2/3	288	192	64-QAM
54	3/4	288	216	64-QAM

* Only supported by the ETSI BRAN HiperLAN2 standard.
** Only supported by the IEEE 802.11a standard.

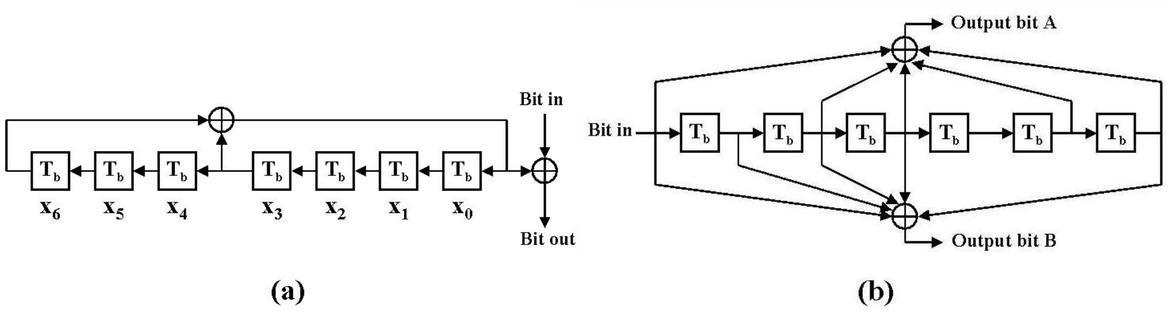


Figure 2.17. Block diagrams: a) Scrambler; b) Convolutional encoder.

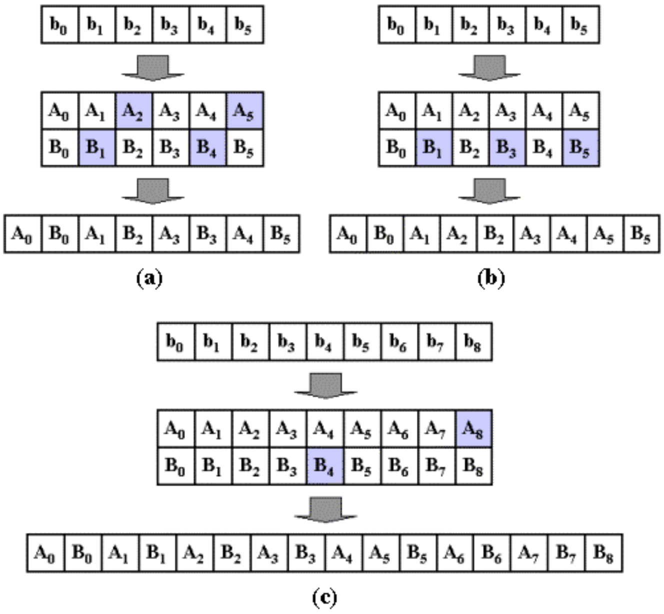


Figure 2.18. "Bit-stealing" algorithm for code rate conversions: a) 1/2 to 2/3; b) 1/2 to 3/4; c) 1/2 to 9/16.

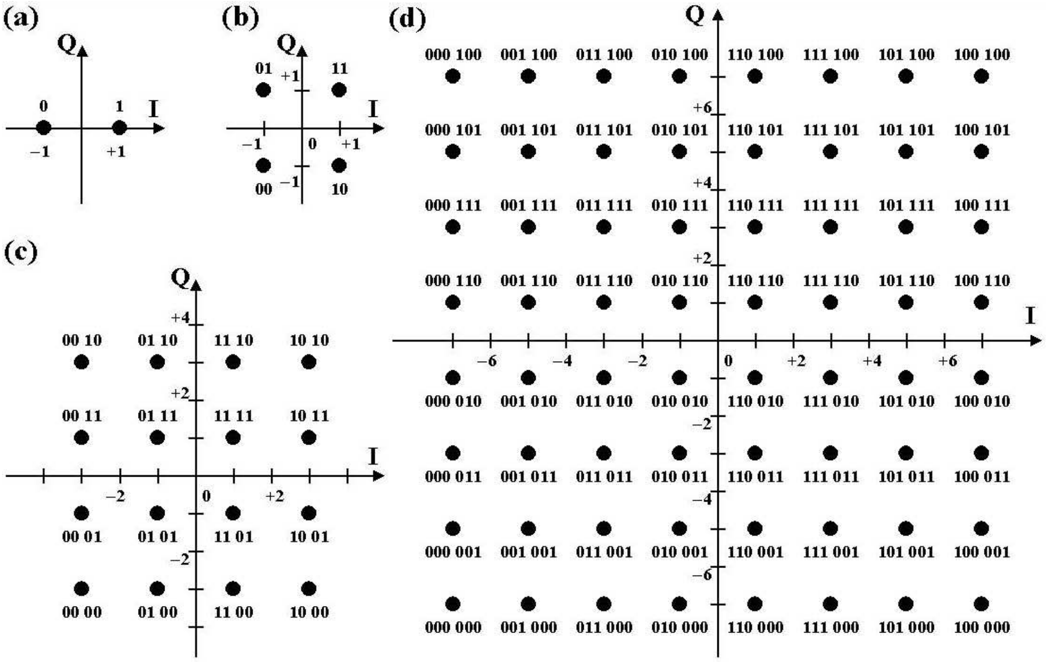


Figure 2.19. Constellation diagram in the considered modulation schemes: a) BPSK; b) QPSK; c) 16-QAM; d) 64-QAM.

2.4.2 Frame Structure

The 802.11a and HL2 standards transmit data in form of frames, i.e. the transmission is not continuous. At the beginning of each frame a group of *preamble symbols* is allocated. The purpose of the preamble is to synchronize the receiver with the incoming frame and to obtain a reference channel estimation. This particular topic is treated in more detail in Chapters 4 and 6, when explaining the Synchronization in OFDM systems. In Figure 2.20 a typical frame structure is shown. Immediately after the preamble symbols, a SIGNAL symbol is transmitted. The information in this symbol is always modulated as in the 6 Mbps case (see Table 2.2), since this is the most robust transmission scheme. The SIGNAL symbol contains 24 bits with information about the transmission rate and the number of bytes being transmitted in that particular frame. The corresponding bit allocation is shown in Figure 2.21. The first four bits (bit 0 to 3) contain the rate information (see [IEEE] and [ETSIa] for more details); the next bit R is reserved for future use and is set to '0'. The bits 5 to 16 contain the number of bytes being transmitted. The maximum value is 4096 octets, i.e. 32,768 bits. The bit P corresponds to the parity bit and is used to perform simple error detection on the first 17 bits. The last six bits (18 to 23) are set to '0' in order to put the convolutional encoder to a known initial state.

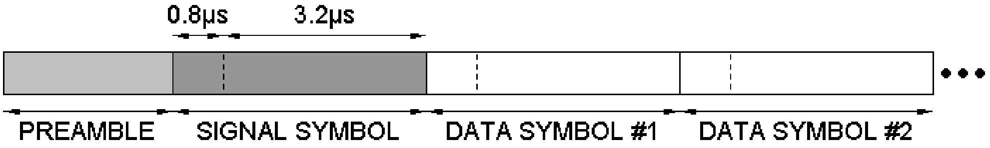


Figure 2.20. Frame structure in the IEEE 802.11a and ETSI HiperlaLAN2 standards.

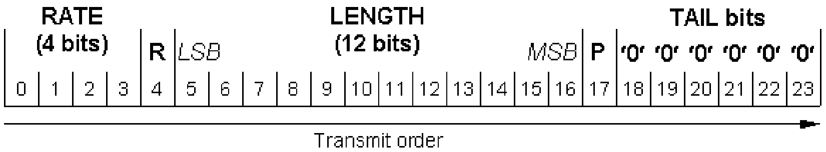


Figure 2.21. Bit content in the SIGNAL symbol.

The DATA symbols shown in Figure 2.20 do not contain directly the user information (payload). Prior and after the payload bits, a number of '0' bits are appended, as shown in Figure 2.22. Concerning the bits prior to the PAYLOAD, it was described in §2.4.1 that the scrambler is initialized randomly on the transmitter side. At the receiver, the scrambler should be also initialized to some state. This is solved by appending seven '0' bits at the very beginning of the bit sequence to be scrambled, as shown in Figure 2.22. These first seven bits will contain the actual state in the scrambler after being scrambled, and therefore, they will be used as the initial state in the scrambler on the receiver side. The next nine bits correspond to the SERVICE bits, which are reserved for future use and are set to '0'.

Six TAIL '0' bits are appended after the PAYLOAD bits before scrambling. After scrambling, the scrambled TAIL bits are replaced by six '0' bits since they are used to set the convolutional encoder to a known state. This procedure was also used within the SIGNAL symbol, in which the last six bits where also defined as TAIL bits. In this manner, the convolutional encoder starts and ends in the same state, thus improving the performance of the decoder on the receiver side.

Regarding the PAD bits in Figure 2.22, they are necessary since the frame should contain an integer number of symbols. As each OFDM symbol contains N_{DBPS} bits (see Table 2.2), the total number of bits in Figure 2.22 should be a multiple of N_{DBPS}. The number of necessary PAD bits, which are inserted before scrambling as '0' bits, is therefore determined by the following expression:

$$N_{PAD} = 1 + \mathit{ceiling}((16 + 8 \cdot \mathit{LENGTH} + 6) / N_{DBPS}), \qquad (2.39)$$

where *LENGTH* states for the number of data bytes transmitted in the frame, and *ceiling*(x) is the nearest integer towards infinity of the real number x. The maximum number of data symbols per frame for the different transmission rates is listed in Table 2.3.

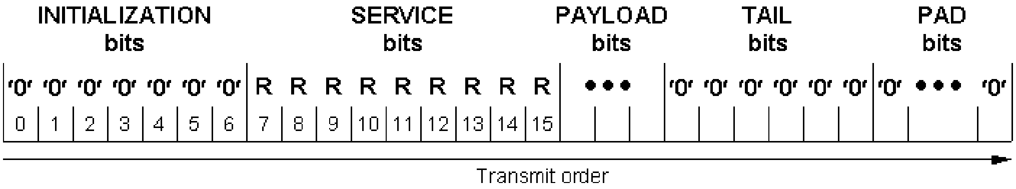


Figure 2.22. Bit content distributed along all the DATA symbols in the frame.

Table 2.3. Maximum number of DATA symbols per frame depending on the nominal transmission rate.

Nominal rate (Mbps)	Maximum frame length (in DATA symbols)
6	1,367
9	911
12	684
18	456
24	342
27	304
36	228
48	171
54	152

2.4.3 Channel Models

In [ETSIb] several channel models for the HiperLAN2 standard are suggested for simulation purposes. The models are also valid for the 802.11a since both systems use the same frequency band. The models in [ETSIb] follow the scheme in Figure 2.5 and provide information about both, the number of taps in the FIR structure and the relative attenuation between the coefficients $g(t,n\mathrm{T})$. The five models are named as A, B, C, D and E, and are provided in a tabular form in Appendix D:

- Model A represented in Table D.1, corresponds to a typical office environment for NLOS conditions and τ_{rms} = 50 ns.
- Model B represented in Table D.2, corresponds to a typical large open space and an office environment for NLOS conditions and τ_{rms} = 100 ns.
- Model C represented in Table D.3, corresponds to a typical large open space indoor or outdoor environment for NLOS conditions and τ_{rms} = 150 ns.
- Model D represented in Table D.4 is the same as model C but for LOS conditions. A 10 dB spike at zero delay is considered, resulting in τ_{rms} = 140 ns.

- Model E represented in Table D.5, corresponds to a typical large open indoor or outdoor space environment for NLOS conditions and τ_{rms} = 250 ns.

In all cases the maximum mobile speed is 3 m/s, i.e. the channel coherence time $T_d \approx 58$ ms, which is much bigger than the symbol time $T_{SYM} = 4$ µs in the reference standards. The Doppler model is the classical Jake's model given in §2.2.2. The results obtained in §2.2.3 are directly applicable here to produce computer simulation models of these channels. In model D, a LOS situation is considered. The Ricean K parameter shown in Table D.4 for the first tap is equivalent to the parameter K_2 in equation (2.7). Furthermore, the parameter $\cos(\gamma)$ in (2.7) has been selected arbitrarily to be 0.7.

Finally, it should be noticed that the maximum delay spread (T_m) allowed by the standards is 750 ns, as mentioned in §2.4.1. On the contrary, channel models C, D and E have power delay profiles longer than this value. In Chapter 5 a solution for this problem will be addressed.

2.5 Summary

In the present Chapter, a simple model for the simulation of wireless channels has been introduced. A number of important parameters have been extracted from this model, namely maximum delay spread (T_m), coherence bandwidth (B_c), Doppler spread (B_d) and coherence time (T_d), which have to be carefully considered when designing OFDM systems. Further on, a mathematical description of the OFDM signals has been provided in which the concept of orthogonality has been analyzed. Note that other more complex functions than the ones provided in (2.22), (2.28) and (2.29) might have been used to achieve orthogonality. Hence, some authors have suggested the use of Wavelet functions in what is called Discrete Wavelet Multitone (DWMT) transmission [Sand95]. Nevertheless, the implementation complexity of such systems is far beyond that of the FFT/IFFT processor found in the classical DMT transmission covered here.

By matrix manipulations it has been found that OFDM signals can cope with multipath channels if a cyclic prefix longer than the maximum delay spread in the channel is provided at each symbol. In this situation, each particular sub-carrier in the OFDM symbol is affected by a multiplicative attenuation factor due to the channel. Consequently, the channel equalization found in single carrier transmissions is greatly simplified in OFDM. Nevertheless, the mobile speed is still a critical issue: since OFDM naturally increases the symbol length, it is also more vulnerable to fast fading.

Finally, two specific standards based on DMT have been described in detail: the IEEE 802.11a and the ETSI BRAN HiperLAN2. Two main topics have been left open during the explanation:

1) The kind of preamble symbols used there. This will be covered in more detail in Chapters 4 and 6, where the synchronization of OFDM signals is discussed.
2) The way to realize channel estimations. This will be covered in more detail in Chapters 5 and 7, where different strategies are investigated.

Next Chapter will analyze the effects of a loss of orthogonality in the OFDM signal, i.e. when expression (2.19) no longer holds at the receiver side.

Chapter 3

Impact of Impairments on OFDM Signals

3.1 Introduction

ORTHOGONALITY IS THE MAIN FEATURE TO BE PRESERVED in an OFDM signal. If this property gets disturbed during transmission and reception, undesirable effects such as ISI and ICI will occur. The aim of this Chapter is the systematic analysis of the most common impairments affecting OFDM signals, as listed by Speth in [Speth99], and to provide specific limits for these impairments in order to decide the degree of accuracy required by the corresponding estimators. Hence the effects of carrier frequency offsets, timing offsets, sampling clock frequency offsets, phase noise, I/Q imbalance and non-linear devices are investigated in the next sections and their impact on the performance is analyzed. All these impairments take place during the RF up- and down-conversion. Some of them are depicted in Figure 3.1 for a typical heterodyne RF down-converter (an RF up-converter would show similar impairments).

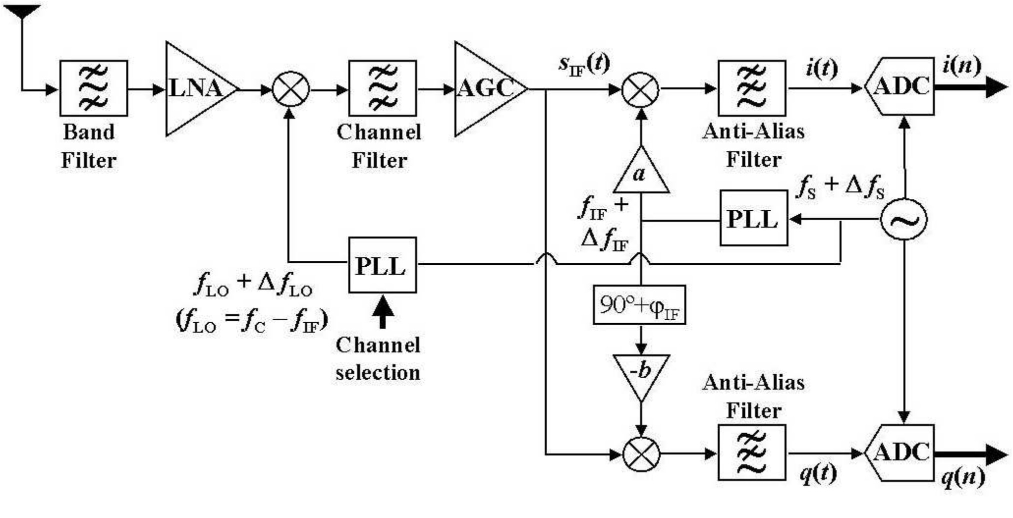


Figure 3.1. Block diagram of a heterodyne RF down-converter showing several impairments.

The scheme in Figure 3.1 considers a commonly used heterodyne architecture in which a single oscillator is present. All the carrier signals necessary for the down-conversion are obtained from this oscillator by using Phase-Locked Loops (PLL). The oscillator is running at a frequency $f_S + \Delta f_S$, where Δf_S represents an offset with respect to the ideal value f_S. In the same way, the RF local oscillator (LO) as well as the intermediate frequency (IF) show an offset given as Δf_{LO} and Δf_{IF}, respectively. In this case it is accomplished that $f_S/\Delta f_S = f_{IF}/\Delta f_{IF} = f_{LO}/\Delta f_{LO}$, since all signals are generated from the same oscillator. The effects derived from a carrier frequency offset on the OFDM signals, i.e. Δf_{LO} and Δf_{IF}, are investigated in §3.2. The offset Δf_S by itself has a different effect on the OFDM signal than that derived from Δf_{LO} and Δf_{IF}, which is investigated in §3.4. The spectral purity of the oscillators, mainly at the output of the PLLs, has to be taken into account as well and is analyzed in §3.5. Yet another effect is the timing offset, which cannot be easily represented in Figure 3.1, but it has a big impact on the demodulation of OFDM signals, as it will be seen in §3.3. The heterodyne receiver in Figure 3.1 generates analog In-phase and Quadrature components, which are further digitized by the ADCs. The effects of mismatches in the I/Q generation, represented here by the parameters a, b and φ_{IF}, are explored in §3.6. The last effect studied in §3.7 relates to the non-linearities found during RF up-conversion due to the power amplifiers and the DACs.

The OFDM signal stream can be expressed as follows considering an infinite number of transmitted symbols,

$$s(t) = \frac{1}{\sqrt{\mathrm{N}}} \sum_{l=-\infty}^{\infty} \left(\sum_{\substack{k=-\mathrm{N_u}/2; \\ (k \neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot e^{j 2\pi \frac{k}{\mathrm{T}} (t - \mathrm{T_G} - l(\mathrm{T}+\mathrm{T_G}))} \right) \cdot w\big(t - l \cdot (\mathrm{T} + \mathrm{T_G})\big), \tag{3.1}$$

where $\mathrm{N_u}$ out of N sub-carriers contain information and the rest $\mathrm{N_v} = \mathrm{N} - \mathrm{N_u}$ are virtual sub-carriers. In addition, the value corresponding to the DC ($k = 0$) is always considered a virtual sub-carrier. The conforming pulse $w(t)$ is taken as a rectangular pulse of length $\mathrm{T} + \mathrm{T_G}$ with the form

$$w(t) = \begin{cases} 1; & 0 \le t \le \mathrm{T} + \mathrm{T_G} \\ 0; & \text{otherwise} \end{cases}, \tag{3.2}$$

where $\mathrm{T_G}$ represents the length of the guard interval or cyclic prefix. The index l indicates the symbol and k stands for the sub-carrier.

The signal $s(t)$ is transmitted over a multipath fading channel as given in §2.2.1

$$h(t,\tau) = \sum_{i=1}^{\mathrm{P}} h_i(t) \cdot \delta(\tau - \tau_i), \tag{3.3}$$

where a fixed number of paths P has been selected, and the coefficients $h_i(t)$ already include the effects of the transmitter filter $g_T(\tau)$ and the receiver filter $g_R(\tau)$.

The received signal $r(t)$ is obtained as the convolution of $s(t)$ and $h(t, \tau)$ plus the addition of AWGN, yielding

$$r(t) = \sum_{i=1}^{\mathrm{P}} h_i(t) \cdot s(t - \tau_i) + v(t)\,. \tag{3.4}$$

On the receiver side, the signal $r(t)$ is sampled at time intervals $t_n = n \cdot (\mathrm{T}/\mathrm{N})$. The channel is considered to be shorter than the cyclic prefix, i.e. $\tau_P < \mathrm{T_G}$, and static for the duration of one OFDM symbol, meaning that the channel is non-time-selective (slow fading). Under these conditions, the signal $r(t_n)$ results as follows

$$r(t_n)=\sum_{i=0}^{\mathrm{P}} h_i(n\frac{\mathrm{T}}{\mathrm{N}})\cdot s(n\frac{\mathrm{T}}{\mathrm{N}}-\tau_i)+v(n\frac{\mathrm{T}}{\mathrm{N}})\,. \tag{3.5}$$

Prior to the DFT calculation, the cyclic prefix is eliminated and only N samples are kept for each symbol. Thus, the relevant samples for further processing are the ones expressed as

$$r_{n,l}=r\left((n+\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G}))\cdot\frac{\mathrm{T}}{\mathrm{N}}\right), \tag{3.6}$$

where $\mathrm{N_G}$ is the extension of the cyclic prefix, i.e. $\mathrm{N_G} = \mathrm{N}\cdot(\mathrm{T_G}/\mathrm{T})$, $l\in(-\infty, +\infty)$ and $0 \le n \le \mathrm{N}-1$.

As shown in §2.3.3, a channel with an impulse response shorter than the cyclic prefix appears as a multiplicative factor inside the OFDM symbols. After performing the DFT, the resulting samples are then as follows

$$Z_{m,l}=\frac{1}{\sqrt{\mathrm{N}}}\sum_{n=0}^{\mathrm{N}-1} r_{n,l}\cdot e^{-j\frac{2\pi}{\mathrm{N}}\cdot n\cdot m}\,, \tag{3.7}$$

or equivalently, including (3.1) into (3.5) yields

$$Z_{m,l}=A_{m,l}\cdot H_{m,l}+V_{m,l}\,, \tag{3.8}$$

where $-\mathrm{N_u}/2 \le m \le \mathrm{N_u}/2$ ($m \ne 0$) and $V_{m,l}$ is complex-valued zero-mean additive white Gaussian noise. The sequence $H_{m,l}$ is obtained as

$$H_{m,l}=\sum_{i=1}^{\mathrm{P}} h_i(l)\cdot e^{-j\frac{2\pi}{\mathrm{T}}\cdot\tau_i\cdot m}\,, \tag{3.9}$$

considering that the channel coefficients $h_i(l)$ are static inside the l-th OFDM symbol.

3.2 Carrier Frequency Offset

The generated OFDM frame $s(t)$ in (3.1) modulates an RF signal with center frequency at f_C. On the receiver side, the signal is initially down-converted using an oscillator tuned at $f_\mathrm{C} \pm f_\varepsilon$, where f_ε represents the total offset in the carrier frequency. Furthermore, there will be an unknown phase offset θ_0 between the oscillators in the transmitter and receiver. The baseband signal is sampled at $f_\mathrm{S} = \mathrm{N}/\mathrm{T}$, so that the samples fed into the DFT block can be expressed as

$$r^{\varepsilon}_{n,l}=e^{j\theta_0}\cdot e^{j\frac{2\pi}{\mathrm{T}}\cdot\varepsilon\cdot\left((n+\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G}))\cdot\frac{\mathrm{T}}{\mathrm{N}}\right)}\cdot r_{n,l}\,, \tag{3.10}$$

where expression (3.6) has been modified accordingly, and $\varepsilon = \mathrm{T}\cdot f_\varepsilon$ is the normalized carrier frequency offset. In this scenario the parameter f_ε is equivalent to $f_\varepsilon = \Delta f_\mathrm{LO}^\mathrm{T} + \Delta f_\mathrm{IF}^\mathrm{T} + \Delta f_\mathrm{LO}^\mathrm{R} + \Delta f_\mathrm{IF}^\mathrm{R}$, where the superscripts $(.)^\mathrm{T}$ and $(.)^\mathrm{R}$ stand for transmitter and receiver, respectively. Furthermore, the problem has been simplified by setting $\Delta f_\mathrm{S} = 0$.

After DFT calculation, the sequence $Z_{m.l}$ results as follows

$$Z_{m,l}=\frac{1}{\sqrt{\mathrm{N}}}\sum_{n=0}^{\mathrm{N}-1} r^{\varepsilon}_{n,l}\cdot e^{-j\frac{2\pi}{\mathrm{N}}\cdot n\cdot m}$$

$$= \frac{e^{j\theta_0}}{\sqrt{\mathrm{N}}} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot\varepsilon\cdot\left(\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G})\right)} \sum_{n=0}^{\mathrm{N}-1} r_{n,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot(\varepsilon-m)\cdot n} \quad , \tag{3.11}$$

with $-\mathrm{N_u}/2 \leq m \leq \mathrm{N_u}/2$ ($m \neq 0$). Considering the results in (3.6), (3.5) and (3.1), the previous expression may be written as

$$\begin{aligned}
Z_{m,l} &= \frac{e^{j\theta_0}}{\mathrm{N}} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot\varepsilon\cdot\left(\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G})\right)} \cdot \left(\sum_{\substack{k=-\mathrm{N_u}/2; \\ (k\neq 0)}}^{\mathrm{N_u}/2} H_{k,l} \cdot A_{k,l} \sum_{n=0}^{\mathrm{N}-1} e^{j\frac{2\pi}{\mathrm{N}}\cdot(k-m+\varepsilon)\cdot n} \right) \\
&= \frac{e^{j\theta_0}}{\mathrm{N}} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot\varepsilon\cdot\left(\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G})\right)} \cdot H_{m,l} \cdot A_{m,l} \cdot e^{j\pi\cdot\varepsilon\cdot\left(\frac{\mathrm{N}-1}{\mathrm{N}}\right)} \cdot \frac{\sin(\pi\varepsilon)}{\sin\left(\frac{\pi}{\mathrm{N}}\varepsilon\right)} \\
&+ \frac{e^{j\theta_0}}{\mathrm{N}} e^{j\frac{2\pi}{\mathrm{N}}\cdot\varepsilon\cdot\left(\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G})\right)} \cdot \left(\sum_{\substack{k=-\mathrm{N_u}/2; \\ (k\neq m; k\neq 0)}}^{\mathrm{N_u}/2} H_{k,l} \cdot A_{k,l} \cdot e^{j\pi(k-m+\varepsilon)\cdot\left(\frac{\mathrm{N}-1}{\mathrm{N}}\right)} \cdot \frac{\sin(\pi(k-m+\varepsilon))}{\sin\left(\frac{\pi}{\mathrm{N}}(k-m+\varepsilon)\right)} \right) \\
&+ V_{m,l} \, ,
\end{aligned} \tag{3.12}$$

where the following well-known expression has been used in the calculation

$$\sum_{n=0}^{\mathrm{N}-1} a^n = \frac{1-a^{\mathrm{N}}}{1-a} \, . \tag{3.13}$$

The first term found in (3.12) for $Z_{m,l}$ represents an attenuated and rotated version of the transmitted data, whereas the second term corresponds to the ICI and the third term is additive Gaussian noise. The phase shift introduced by the frequency offset does not depend on m, thus being constant inside the OFDM symbol, but it gets accumulated from symbol to symbol, since it includes the index l.

Our main goal in this analysis is the determination of the maximum acceptable frequency offset necessary to achieve a certain Interference-to-Signal Ratio (ISR). Statistically, the ICI term in (3.12) is modeled as Gaussian noise since it is the result of an addition of $\mathrm{N_u}$ random variables (Central Limit Theorem), where $\mathrm{N_u}$ is generally large. Considering that the channel is normalized and statistically independent of the transmitted information, and that the information transmitted into different sub-carriers is uncorrelated with each other but all have the same power σ_A^2, the ICI power affecting sub-carrier m is given by

$$\sigma_\mathrm{I}^2(m) = \sigma_\mathrm{A}^2 \frac{1}{\mathrm{N}^2} \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq m; k\neq 0)}}^{\mathrm{N_u}/2} \frac{\sin^2(\pi(k-m+\varepsilon))}{\sin^2\left(\frac{\pi}{\mathrm{N}}(k-m+\varepsilon)\right)} \, , \tag{3.14}$$

with $-\mathrm{N_u}/2 \leq m \leq \mathrm{N_u}/2$ ($m \neq 0$).

In Figure 3.2 the ratio ($\sigma_\mathrm{I}^2 / \sigma_\mathrm{A}^2$), i.e. the Interference-to-Signal Ratio (ISR), is represented as a function of the normalized frequency offset ε for the case of $\mathrm{N} = 64$, $\mathrm{N_u} = 52$ and for two extreme values of m, i.e. $m = 1$ and $m = 26$. The resulting curves show little variation of the ISR with respect to the sub-carrier value m. As a design parameter, it should be decided the allowed degradation in the SNR per sub-carrier and therefore the maximum error in the estimation and correction of the frequency offset. As a rule of thumb, we arbitrarily selected a value for the ISR as small as –40dB. Hence, according to Figure 3.2, the normalized frequency offset should be below $\varepsilon = 0.01$.

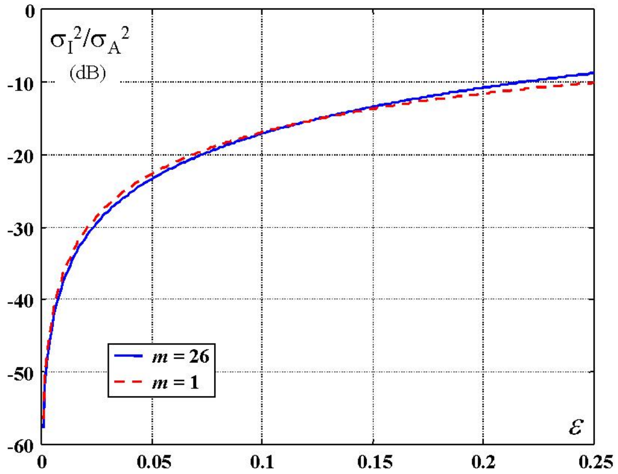


Figure 3.2. Interference-to-Signal Ratio as a function of the normalized frequency offset (ε), for the case of N=64, N_u=52.

3.3 Timing Offset

Let's consider now the case when the starting point of the frame $s(t)$ is not well determined, but it is affected by a timing error $t_\theta = \theta \cdot (T/N)$. The signal $s(t)$ is sampled at instants

$$n' = n + \theta + N_G + l' \cdot (N + N_G). \tag{3.15}$$

Equivalently, expression (3.6) is modified as follows

$$r_{n,l'}^{\theta} = r\left((n + \theta + N_G + l' \cdot (N + N_G)) \cdot \frac{T}{N} \right), \tag{3.16}$$

where a new symbol index l' is taken into account on the receiver side. Inserting (3.4) into (3.16) yields

$$r_{n,l'}^{\theta} = \frac{1}{\sqrt{N}} \sum_{\substack{k=-N_u/2 \\ (k \neq 0)}}^{N_u/2} A_{k,l} \cdot e^{j\frac{2\pi}{N}k\left(n+\theta+(l'-l)\cdot(N+N_G)\right)} \cdot w'\left((n + \theta + N_G + l' \cdot (N + N_G)), l\right) + v(n'), \tag{3.17}$$

with $w'(n, l)$ given by the following expression,

$$w'(n,l) = \sum_{i=1}^{P} h_i(l) \cdot e^{-j\frac{2\pi}{T}\tau_i} \cdot w\left(n - \tau_i \cdot (N/T) - l \cdot (N + N_G)\right), \tag{3.18}$$

considering that the channel is approximately constant inside the OFDM symbol. The transmission will be undisturbed if the samples belonging to symbol l do not influence those belonging to symbol l'. To find out the possible values of θ accomplishing this condition, we should remember the definition of $w(t)$ given in (3.2). For the case of a sampled window, this yields

$$w(n)=\begin{cases}1; & 0\le n<\mathrm{N}+\mathrm{N_G}\\ 0; & \text{otherwise}\end{cases}. \tag{3.19}$$

The received pulse $w'(n, l)$ in (3.18) represents a weighted addition of P delayed pulses. This results in an extension of the pulse by τ_P, where τ_P stands for the longest path in the channel impulse response. In general, the ratio $(\mathrm{N/T})\cdot\tau_P$ is not going to be an integer number. Instead, the value $\mathrm{N_m} = ceiling\,((\mathrm{N/T})\cdot\tau_P)$ is taken for calculation, where the function *ceiling* (x) represents the nearest integer towards infinity of x. Furthermore, the condition $0 \le \mathrm{N_m} \le \mathrm{N_G}$ should be satisfied in order to avoid ISI (see Appendix C).

Under these circumstances, the condition

$$w'\left(n+\theta+\mathrm{N_G}+l'\cdot(\mathrm{N}+\mathrm{N_G}),l\right)=0 \tag{3.20}$$

must be met in order to avoid ISI, considering the two neighboring symbols of l, i.e. $l'-l = \Delta l = \pm1$, and for all the possible values of the FFT window, $0 \le n \le \mathrm{N} - 1$. In Figure 3.3, the window $w'(n + \mathrm{N_G} + l'\cdot(\mathrm{N}+\mathrm{N_G}), l)$ is represented for the three cases of interest, i.e. $\Delta l = -1, 0, +1$. The shadowed regions symbolize the extension suffered by $w(n)$ due to the multipath channel. In order to avoid ISI, no piece of information of the neighboring symbols should fall inside the FFT window of the symbol being processed, $0 \le n \le \mathrm{N} - 1$ ($l = 0$), when a value $\theta \neq 0$ is considered. For the window at $\Delta l = -1$ ($l = 1$) this is only possible if $\theta \le 0$, whereas for $\Delta l = +1$ ($l = -1$) this is possible if $\theta \ge \mathrm{N_m}-\mathrm{N_G}$, yielding

$$\mathrm{N_m}-\mathrm{N_G}\le\theta\le 0\,. \tag{3.21}$$

The condition (3.21) states that there is no single possible starting point for the OFDM symbols. Instead, the symbol starting may be taken at any point in the region where the cyclic prefix is not affected by the ISI.

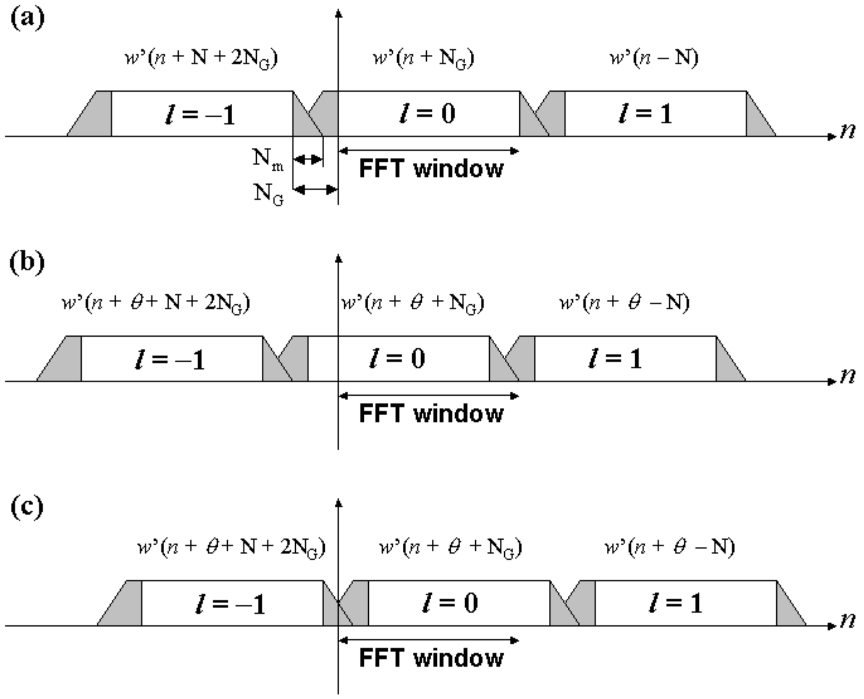


Figure 3.3. Inter-Symbol Interference depending on θ for: a) $\theta = 0$; b) $\theta > 0$; c) $\theta < (\mathrm{N_m} - \mathrm{N_G})$. To avoid ISI, no samples corresponding to symbols $l = -1$ or $l = +1$ should fall into the FFT window of symbol $l = 0$.

Appendix E contains the mathematical derivation of the ISI and ICI components generated after FFT when the condition in (3.21) is not fully satisfied. The analysis considers the two possible situations that may arise, i.e. $\theta > 0$ and $\theta < \mathrm{N_m}-\mathrm{N_G}$ ($\theta < 0$). Nevertheless, the worst case occurs when $\theta > 0$, resulting in the following sequence after FFT calculation,

$$Z_{m,l} = A_{m,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}m\cdot\theta} \cdot \left(H_{m,l} - H^{\theta}_{m,l}\right) + \sum_{\substack{k=-\mathrm{N_u}/2;\\(k\neq m;k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot \alpha((k-m),\theta) \cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot\theta} + Z^{\mathrm{ISI}}_{m,l'}\Big|_{\Delta l=-1} + v(n')\,, \tag{3.22}$$

with $-\mathrm{N_u}/2 \leq m \leq \mathrm{N_u}/2$ ($m \neq 0$) and $H^{\theta}_{m,l}$ being given as in (E.9). The ISI component found in (3.22) is given as in (E.11). Four impairments may be distinguished in (3.22):

1) The expected signal $A_{m,l}$ will be rotated by a linear phasor depending on θ. This phasor does not depend on l, thus remaining constant from symbol to symbol.
2) The channel "seen" by the receiver is no longer $H_{m,l}$ but $H_{m,l} - H^{\theta}_{m,l}$, leading to errors in the channel estimation.
3) There will be an ICI component (second summand) derived from the loss of orthogonality. As explained in §3.2.1 for the case of the carrier frequency offset, the ICI is also modeled as Gaussian noise (CLT) and is statistically independent of $A_{m,l}$ and the AWGN.
4) The ISI component (third term) is generated by the samples of the next symbol (symbol $l = 1$ in Figure 3.3b) that fall into the FFT window. It is also modeled as Gaussian noise and is statistically independent of $A_{m,l}$ and the AWGN.

In Figure 3.4 the Interference-to-Signal Ratio $(\sigma_{\mathrm{ICI}}^2 + \sigma_{\mathrm{ISI}}^2) / \sigma_{\mathrm{A}}^2$ is represented as a function of a positive timing offset for the case of an AWGN channel and a multipath channel model A with normalized power, as given in Table D.1.

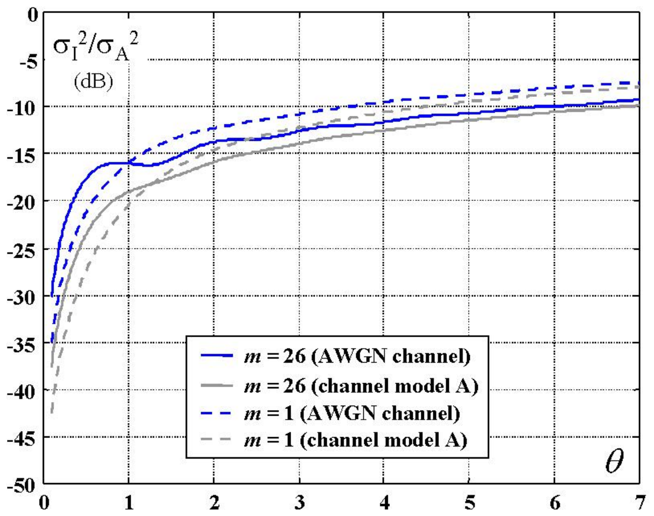


Figure 3.4. Interference-to-Signal Ratio as a function of the normalized timing offset θ (N = 64, $\mathrm{N_u}$ = 52).

In a sampled system, a residual error in the timing estimation will be always present. This error is uniformly distributed in the range $\pm T_S/2$, where T_S represents the sampling time. According to Figure 3.5, a value of $\theta = 0.5$ (normalized with respect to T_S) already creates a big ISR, higher than –20 dB in some sub-carriers. Since θ is expressed relative to the sampling time, a possible way to reduce it would be by increasing the sampling frequency during synchronization. Nevertheless, this has a major disadvantage in the fact that the whole digital circuitry has to be run at higher frequencies, increasing the implementation complexity. An alternative solution based on variable interpolation will be presented later on in Chapter 7.

3.4 Sampling Clock Frequency Offset

This problem arises when the sampling interval at the receiver is not exactly $T_S = T/N$ (or a multiple of it), but it is affected by an error in the form $T_S' = (1+\zeta)\cdot(T/N)$. The parameter ζ may be positive or negative depending on whether the sampling is slower or faster than expected, respectively. Similarly as it was done for the case of the timing offset, the samples at the receiver side may be expressed as

$$n' = (n + N_G + l'\cdot(N + N_G))\cdot(1+\zeta) = n + n_\zeta(n, l') + N_G + l'\cdot(N + N_G), \tag{3.23a}$$

$$n_\zeta(n, l') = \zeta\cdot(n + N_G + l'\cdot(N + N_G)), \tag{3.23b}$$

where l' is the symbol index at the receiver side. The parameter ζ is expressed in parts-per-million (ppm) and generally ranges from 20 to 100 ppm, i.e. $2\cdot10^{-5}$ to 10^{-4}.

From expression (3.23a) it is straightforward to see that the effect of a sampling clock frequency offset is a *variable timing offset*. This means that not only a linear phase error will be visible inside each OFDM symbol, as shown in (3.22), but also the slope of this linear phase error will change on a symbol basis.

As explained in §3.3, a timing offset would originate both ICI and ISI components in addition to the linear phase error if the condition $-N_G+N_m \le n_\zeta(n, l') \le 0$ is not satisfied. Considering (3.17), with $l'-l=0$ and $\theta = n_\zeta(n, l')$, the signal after DFT results as follows

$$Z_{m,l} = \frac{1}{N} \sum_{\substack{k=-N_u/2 \\ (k\neq 0)}}^{N_u/2} A_{k,l} \cdot e^{j\frac{2\pi}{N}k\cdot\zeta\cdot(N_G + l\cdot(N+N_G))} \left(\sum_{i=1}^{P} h_i(l)\cdot e^{-j\frac{2\pi}{T}k\tau_i} \cdot W_i((k-m),\zeta,l) \right) + v(n'), \tag{3.24}$$

with $-N_u/2 \le m \le N_u/2$ $(m \neq 0)$ and

$$W_i((k-m),\zeta,l) = \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k\cdot(1+\zeta)-m)\cdot n} w\big((1+\zeta)\cdot(n+N_G) - \tau_i\cdot(N/T) + \zeta\cdot l\cdot(N+N_G)\big). \tag{3.25}$$

By defining

$$\alpha((k-m),\zeta,l) = \frac{1}{N}\sum_{i=1}^{P} h_i(l)\cdot e^{-j\frac{2\pi}{T}k\cdot\tau_i} \cdot W_i((k-m),\zeta,l), \tag{3.26}$$

the expression (3.24) results as follows,

$$Z_{m,l} = A_{m,l}\cdot\alpha(0,\zeta,l)\cdot e^{j\frac{2\pi}{N}m\cdot\zeta\cdot(N_G+l\cdot(N+N_G))} + \sum_{\substack{k=-N_u/2 \\ (k\neq m; k\neq 0)}}^{N_u/2} A_{k,l}\cdot e^{j\frac{2\pi}{N}k\cdot\zeta\cdot(N_G+l\cdot(N+N_G))} \cdot\alpha((k-m),\zeta,l) + v(n') \tag{3.27}$$

The analysis of the function $\alpha((k-m), \zeta, l)$ in (3.26) is not straightforward since the effect of the parameter ζ on the window $w(n)$ in (3.25) is double. On one hand, it represents an expansion or compression of $w(n)$ by a factor $(1+\zeta)^{-1}$, depending on whether ζ is negative or positive, respectively. On the other side, ζ introduces a delay, which depends on the particular symbol l. Nevertheless, the factor $(1+\zeta)^{-1}$ may be approximated by 1 since it is a constant value throughout the symbols and ζ is typically very small. Then, (3.25) results in an expression very similar to (E.3), with θ being $\zeta \cdot l \cdot (\mathrm{N}+\mathrm{N}_\mathrm{G})$. For the first symbols the approximation $\alpha(0, \zeta, l) \approx H_{m,l}$ will hold, but as l increases, the channel mismatch will become much more evident.

When $\zeta > 0$, i.e. sampling frequency slower than expected, ISI and ICI factors will appear in the data after DFT. To have an idea of the impact of ζ, let's consider the case of an OFDM system based on a frame transmission, as it is the case of the 802.11a and HL2 standards. The values of l will be restricted to be in the range $[0, \mathrm{S}-1]$, where S is the maximum number of OFDM symbols transmitted inside one frame. For the particular case of these standards, the worst case is the transmission in the 6 Mbps mode (see Table 2.3), being $\mathrm{S} \approx 1{,}370$ symbols. The standards also allow a value for ζ in the range 20 ppm [IEEE], [ETSIa], i.e. $\zeta = 2 \cdot 10^{-5}$. The product $\zeta \cdot (\mathrm{S}-1) \cdot (\mathrm{N}+\mathrm{N}_\mathrm{G})$, with $\mathrm{N}+\mathrm{N}_\mathrm{G}=80$, is about 2.2 samples, which represents a very big timing offset and cannot be left uncorrected. In the 54 Mbps case, $\mathrm{S} \approx 155$ and the maximum timing error is 0.25 samples, but since an extremely sensitive modulation scheme is used, i.e. 64-QAM, the previous timing error would be enough to induce unreliable demodulation.

If $\zeta < 0$, no ICI or ISI will occur as long as $-\mathrm{N}_\mathrm{G}+\mathrm{N}_\mathrm{m} \leq \zeta \cdot (\mathrm{S}-1) \cdot (\mathrm{N}+\mathrm{N}_\mathrm{G})$, considering an analogy to the results obtained for the timing offset in §3.3. Hence, the condition $\zeta < 0$ is more preferable and should be forced during tracking mode. A possible solution will be thoroughly investigated in Chapter 7.

3.5 Phase Noise Modeling

The RF oscillators found in the AFE (see Figure 3.1) introduce other impairments rather than a carrier frequency offset. Actual oscillators are not able to generate pure spectral components, but these components show spectral “tails” called *phase noise*. Hence, the output of an oscillator may be generally expressed as

$$e^{j(2\pi \cdot f_0 \cdot t + \varphi_N(t))} \tag{3.28}$$

where f_0 is a generic oscillation frequency and $\varphi_N(t)$ represents a realization of the phase noise process, with $\varphi_N(t) << 1/(2\pi)$ $\forall t$. The analysis of the effects of the phase noise on OFDM signals was firstly introduced in [Rob95] and [Mus95].

Considering (3.28), the sampled baseband OFDM signal prior to DFT will be given as follows,

$$r_{n,l}^{\varphi} = e^{j\varphi_N((n+\mathrm{N}_\mathrm{G}+l\cdot(\mathrm{N}+\mathrm{N}_\mathrm{G})))} \cdot r_{n,l} \approx (1 + j\varphi_N(n+\mathrm{N}_\mathrm{G}+l\cdot(\mathrm{N}+\mathrm{N}_\mathrm{G}))) \cdot r_{n,l}, \tag{3.29}$$

and after DFT, yields

$$Z_{m,l} = \frac{1}{\sqrt{\mathrm{N}}} \sum_{n=0}^{\mathrm{N}-1} r_{n,l}^{\varphi} \cdot e^{-j\frac{2\pi}{\mathrm{N}} \cdot n \cdot m}$$

$$= A_{m,l} \cdot H_{m,l} + j\frac{1}{\sqrt{\mathrm{N}}}\sum_{n=0}^{\mathrm{N}-1}\varphi_N\left(n+\mathrm{N}_\mathrm{G}+l(\mathrm{N}+\mathrm{N}_\mathrm{G})\right)\cdot r_{n,l}\cdot e^{-j\frac{2\pi}{\mathrm{N}}\cdot n\cdot m}$$

$$= A_{m,l} \cdot H_{m,l} + \frac{j}{\mathrm{N}}\sum_{\substack{k=-\mathrm{N_u}/2\\(k\neq 0)}}^{\mathrm{N_u}/2} H_{k,l}\cdot A_{k,l}\sum_{n=0}^{\mathrm{N}-1}\varphi_N\left(n+\mathrm{N}_\mathrm{G}+l(\mathrm{N}+\mathrm{N}_\mathrm{G})\right)\cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot(k-m)\cdot n}$$

$$= e^{j\psi_N(l)} H_{m,l}\cdot A_{m,l} + \frac{j}{\mathrm{N}}\sum_{\substack{k=-\mathrm{N_u}/2\\(k\neq m;k\neq 0)}}^{\mathrm{N_u}/2} H_{k,l}\cdot A_{k,l}\sum_{n=0}^{\mathrm{N}-1}\varphi_N\left(n+\mathrm{N}_\mathrm{G}+l(\mathrm{N}+\mathrm{N}_\mathrm{G})\right)\cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot(k-m)\cdot n}, \qquad (3.30)$$

with $-\mathrm{N_u}/2 \le m \le \mathrm{N_u}/2$ $(m \neq 0)$ and

$$e^{j\psi_N(l)} = 1+\frac{j}{\mathrm{N}}\sum_{n=0}^{\mathrm{N}-1}\varphi_N\left(n+\mathrm{N}_\mathrm{G}+l(\mathrm{N}+\mathrm{N}_\mathrm{G})\right). \qquad (3.31)$$

The phase in (3.31) represents a Common Phase Error (CPE) [Rob95], which is constant inside one OFDM symbol, but changes from symbol to symbol in a *random* way. Furthermore, from (3.30) an ICI component is added to the signal of interest due to the lost of orthogonality. In (3.31) a pure phasor is considered since $\varphi_N(t) << 1$. The phase $\psi_N(l)$ is obtained as

$$\psi_N(l) = \arg\left\{1+\frac{j}{\mathrm{N}}\sum_{n=0}^{\mathrm{N}-1}\varphi_N\left(n+\mathrm{N}_\mathrm{G}+l(\mathrm{N}+\mathrm{N}_\mathrm{G})\right)\right\} \approx \frac{1}{\mathrm{N}}\sum_{n=0}^{\mathrm{N}-1}\varphi_N\left(n+\mathrm{N}_\mathrm{G}+l(\mathrm{N}+\mathrm{N}_\mathrm{G})\right), \qquad (3.32)$$

where $\psi_N(l)$ can be very well approximated by a zero-mean Gaussian process (Central Limit Theorem). If the bandwidth of the process $\varphi_N(t)$ is much less than the sub-carrier spacing inside the OFDM symbol, Δf, the expression (3.32) may be further simplified as follows

$$\psi_N(l) \approx \frac{1}{\mathrm{T}}\int_{(\mathrm{T}+\mathrm{T_G})\cdot l+\mathrm{T_G}}^{(\mathrm{T}+\mathrm{T_G})\cdot(l+1)}\varphi_N(x)\,dx\,. \qquad (3.33)$$

In the analysis made in §§3.1, 3.2 and 3.3, all the impairments showed a linear phase contribution in the data $A_{m,l}$. The overall phase contribution can be easily estimated in those cases since it is a deterministic value. In the case of the CPE, the autocorrelation function (ACF) should be found in order to determine whether the phases $\psi_N(l)$ are very much correlated between adjacent symbols. This ACF is obtained as the inverse Fourier Transform of the Power Density Spectrum (PDS) of the random process. Because the phase noise process is stationary, the PDS of the CPE can be written as in (3.34), considering that the integration interval in (3.33) is of length T, i.e.

$$\Psi_N(f) \approx \Phi_N(f)\cdot\left|\frac{\sin(\pi \mathrm{T} f)}{\pi \mathrm{T} f}\right|^2. \qquad (3.34)$$

Therefore, the ACF may be expressed as follows,

$$\mathrm{R}_{\psi_N}(t) = \int_{-\infty}^{\infty}\Phi_N(f)\cdot\left|\frac{\sin(\pi \mathrm{T} f)}{\pi \mathrm{T} f}\right|^2\cdot e^{j2\pi\cdot f\cdot t}df\,. \qquad (3.35)$$

Here a model for the PDS of the phase noise is needed. In [Bahai99] the authors suggest a Lorentzian model for the phase noise generated by a phase-locked oscillator (PLL) with the form

$$\Phi_N(f) = \frac{1}{\pi}\frac{\mathrm{K}_\Phi \cdot \beta_\Phi}{f^2 + \beta_\Phi^2};\ -\infty < f < \infty, \tag{3.36}$$

where K_Φ is the total power of the phase noise, and β_Φ is the 3 dB bandwidth. In order to solve the integral in (3.35) it is considered that $\beta_\Phi << \Delta f = 1/\mathrm{T}$, thus making $\mathrm{sinc}^2(f \cdot \mathrm{T}) \approx 1$. Under this condition, (3.35) results as follows

$$\mathrm{R}_{\psi_N}(t) = \frac{2}{\pi}\int_0^\infty \frac{\mathrm{K}_\Phi \cdot \beta_\Phi}{f^2 + \beta_\Phi^2} \cdot \cos(2\pi \cdot f \cdot t)\, df\ . \tag{3.37}$$

The previous integral is tabulated in [Bron74] as

$$\int_0^\infty \frac{\cos(nx)}{1+x^2}\, dx = \frac{\pi}{2} e^{-|n|}, \tag{3.38}$$

thus providing the following result for the ACF,

$$\mathrm{R}_{\psi_N}(t) = \mathrm{K}_\Phi \cdot e^{-|2\pi\beta_\Phi \cdot t|}\ . \tag{3.39}$$

However, our interest is centered on the value of the ACF at those instants $t = l \cdot (\mathrm{T}+\mathrm{T}_\mathrm{G})$, yielding

$$\mathrm{R}_{\psi_N}(l) = \mathrm{K}_\Phi \cdot e^{-\left|2\pi\beta_\Phi \cdot l\mathrm{T}\left(1+\frac{\mathrm{N_G}}{\mathrm{N}}\right)\right|}\ . \tag{3.40}$$

The condition $\beta_\Phi << \Delta f$ may not be true for certain values of N, thus making the equation in (3.40) of little use. The symbol timing is given as $\mathrm{T} = \mathrm{N}/\mathrm{B_W}$, where $\mathrm{B_W}$ is the total bandwidth of the OFDM signal. If $\mathrm{B_W}$ is kept constant and N is increased, then Δf decreases and a simplification of the integral in (3.35) is no longer possible. Therefore, this integral has to be solved analytically. Equivalently, according to (3.40), the variance of the CPE is given as $\mathrm{R}_{\psi N}(0) = \mathrm{K}_\Phi$, but this will only stand as long as $\beta_\Phi << \Delta f$.

The normalized variance of the CPE (with respect to K_Φ) has been represented in Figure 3.5 as a function of the number of sub-carriers N. Immaterial to the value of β_Φ, the variance decreases as N is increased. Furthermore, the bigger is the noise bandwidth, the more rapidly decreases the variance. Hence, systems making use of a small number of sub-carriers, e.g. HiperLAN2, will be much more affected by the CPE than others, e.g. DVB, where 8,192 tones are employed.

Similarly, the normalized ACF of the CPE is depicted in Figure 3.6 for different values of N. It can be seen that for smaller FFT sizes, the correlation of the CPE between adjacent OFDM symbols increases, which may simplify the implementation of a CPE estimation mechanism [Rob95].

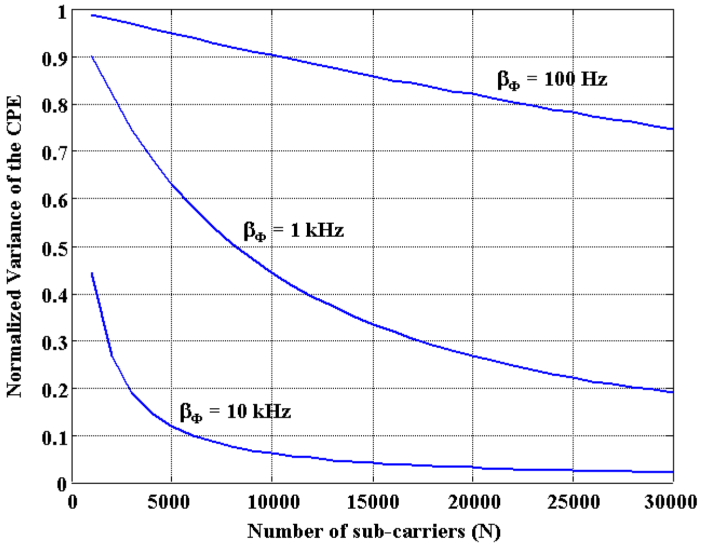


Figure 3.5. Normalized variance of the CPE (in radians) with $B_W = 20$ MHz.

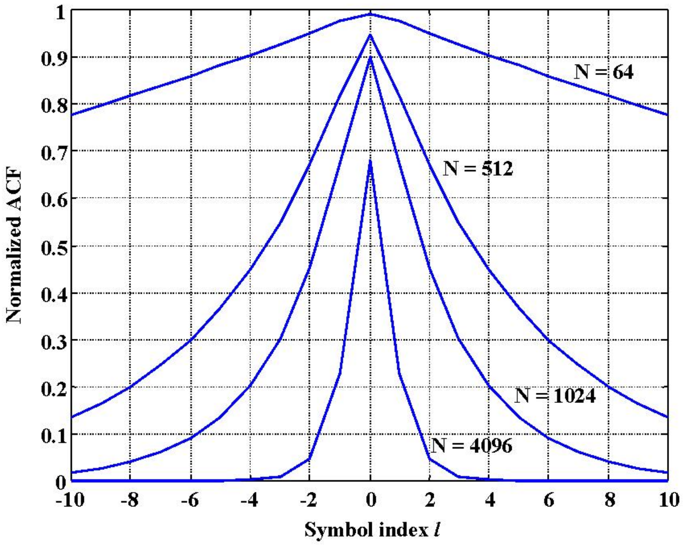


Figure 3.6. Normalized ACF of the CPE for $\beta_\Phi = 1$ kHz ($B_W = 20$ MHz).

The second term to be discussed here is the ICI component appearing in (3.30), which is given by the following expression,

$$\mathrm{I} = \frac{j}{\mathrm{N}} \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq m;k\neq 0)}}^{\mathrm{N_u}/2} H_{k,l} \cdot A_{k,l} \sum_{n=0}^{\mathrm{N}-1} \varphi_N(n+\mathrm{N_G}+l(\mathrm{N}+\mathrm{N_G})) \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot(k-m)\cdot n} . \qquad (3.41)$$

The power of the ICI component as a function of sub-carrier m can be given as in (3.42), considering a normalized channel independent of the transmitted information $A_{k,l}$. Furthermore, the information transmitted onto different sub-carriers is independent with each other, but all them have the same power σ_A^2, yielding

$$\sigma_{\mathrm{I}}^2(m) = \sigma_{\mathrm{A}}^2 \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq m;k\neq 0)}}^{\mathrm{N_u}/2} \int_{-\infty}^{\infty} \Phi_N(f) \cdot \left| \frac{\sin(\pi(k-m+\mathrm{T}f))}{\pi(k-m+\mathrm{T}f)} \right|^2 df , \qquad (3.42)$$

with $-\mathrm{N_u}/2 \le m \le \mathrm{N_u}/2$ ($m \neq 0$).

The result of this integral is represented in Figure 3.7 normalized with respect to K_Φ for two extreme values of m, in the case of the 802.11a and HL2 standards. From Figure 3.7 it can be observed that for a fixed value of β_Φ and B_W, the ICI power increases as the number of sub-carriers N is increased (Δf is decreased). In the specific case of the previously mentioned standards, if β_Φ = 1 kHz, then the ratio $\beta_\Phi / \Delta f = 0.0032$, which results in very small ICI. On the contrary, systems such as DVB, with B_W = 8 MHz and up to 8,192 sub-carriers, result in $\Delta f \approx 977$ Hz and $\beta_\Phi / \Delta f = 1.024$. In this case, the phase noise bandwidth should be designed to be well below 10 Hz.

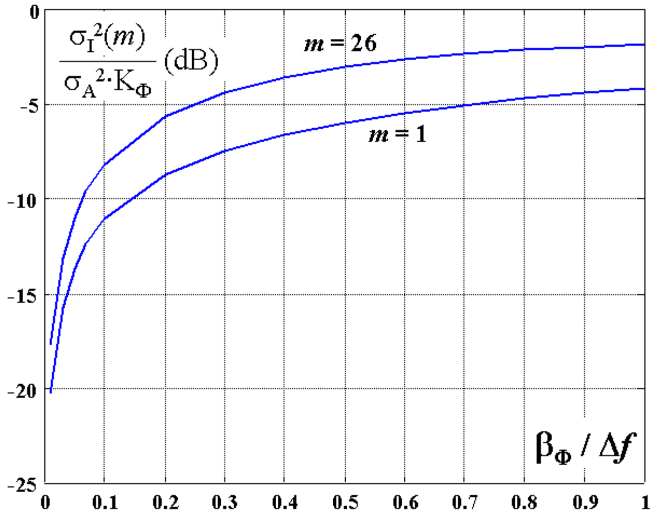


Figure 3.7. Normalized Interference-to-Signal Ratio as a function of the ratio $\beta_\Phi / \Delta f$.

3.6 In-phase and Quadrature Mismatch Modeling

The signal $s(t)$ in (3.1) represents a *low-pass* signal with real and imaginary parts given as $s_I(t)$ and $s_Q(t)$, respectively. In a typical heterodyne receiver, as the one shown in Figure 3.1, the input signal is initially down-converted to an intermediate frequency (f_{IF}), thus obtaining the signal $s_{IF}(t)$. Afterwards the In-phase (I) and Quadrature (Q) components are derived from $s_{IF}(t)$ by multiplying this signal by a cosine and sine function, respectively. At this point, it is of major importance to know the effects of an imbalance between the I and Q branches in Figure 3.1. The authors in [Ben96] firstly analyzed the problem by considering mismatches in the frequency response of the two Anti-Alias Filters (AAF) found in Figure 3.1.

In our case, the I/Q mismatch is studied under another perspective. Hence, the problem is restricted to the analysis of a mismatch in the generation of the cosine and sine functions in both transmitter and receiver side, as shown in Figure 3.8. The low-pass filters before the Analog-to-Digital and after the Digital-to-Analog conversions are considered to be completely symmetrical.

From (3.1), the In-phase and Quadrature components of $s(t)$ may be expressed as

$$s_{\mathrm{I}}(t) = \frac{1}{\sqrt{\mathrm{N}}}\left(\sum_{k=0}^{\mathrm{N}-1} I_{k,l} \cdot \cos\left(\frac{2\pi}{\mathrm{T}} k\left(t - \mathrm{T_G} - l(\mathrm{T} + \mathrm{T_G})\right)\right) - Q_{k,l} \cdot \sin\left(\frac{2\pi}{\mathrm{T}} k\left(t - \mathrm{T_G} - l(\mathrm{T} + \mathrm{T_G})\right)\right)\right)$$

$$\times w\big(t - l\cdot(\mathrm{T}+\mathrm{T_G})\big), \tag{3.43a}$$

$$s_{\mathrm{Q}}(t) = \frac{1}{\sqrt{\mathrm{N}}}\left(\sum_{k=0}^{\mathrm{N}-1} I_{k,l}\cdot\sin\!\left(\frac{2\pi}{\mathrm{T}}k\big(t-\mathrm{T_G}-l(\mathrm{T}+\mathrm{T_G})\big)\right) + Q_{k,l}\cdot\cos\!\left(\frac{2\pi}{\mathrm{T}}k\big(t-\mathrm{T_G}-l(\mathrm{T}+\mathrm{T_G})\big)\right)\right)$$

$$\times w\big(t - l\cdot(\mathrm{T}+\mathrm{T_G})\big), \tag{3.43b}$$

where it has been considered that $A_{k,l} = I_{k,l} + j\cdot Q_{k,l}$ ($j = \sqrt{-1}$), and the index k ranges from 0 to $\mathrm{N} - 1$. Furthermore, no noise has been included in order to simplify the analysis.

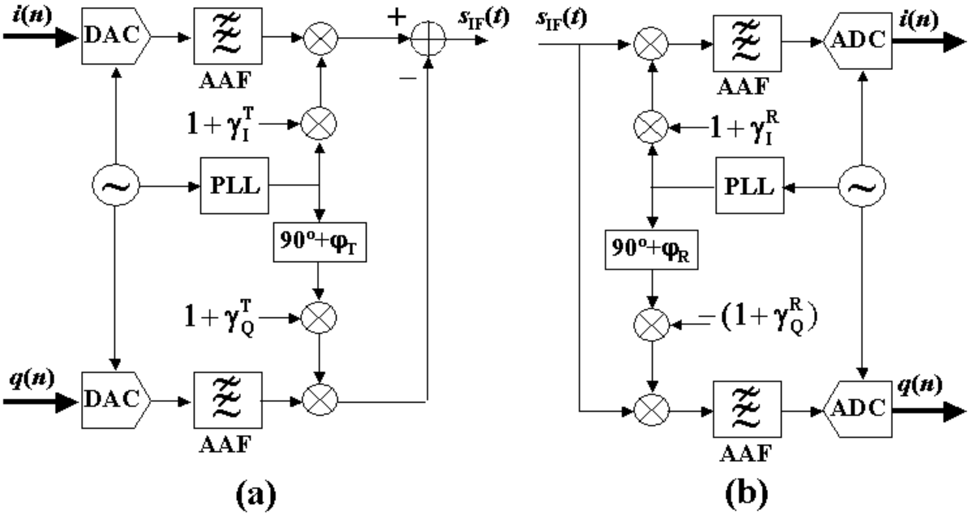


Figure 3.8. Model for the I/Q mismatch: a) Transmitter side; b) Receiver side.

According to the model in Figure 3.8, six parameters are used to represent the I/Q mismatch: $\gamma_{\mathrm{I}}^{\mathrm{T}}$, $\gamma_{\mathrm{Q}}^{\mathrm{T}}$, $\gamma_{\mathrm{I}}^{\mathrm{R}}$, $\gamma_{\mathrm{Q}}^{\mathrm{R}}$, φ_{T} and φ_{R}. Therefore, after transmission and reception of the signal $s(t)$ in (3.1), the received I and Q components may be written as follows,

$$r_{\mathrm{I}}(t) = \frac{1}{2}(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})\cdot s_{\mathrm{I}}(t) - \frac{1}{2}\sin(\varphi_{\mathrm{T}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})\cdot s_{\mathrm{Q}}(t), \tag{3.44a}$$

$$r_{\mathrm{Q}}(t) = \frac{1}{2}\cos(\varphi_{\mathrm{T}}-\varphi_{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})\cdot s_{\mathrm{I}}(t) - \frac{1}{2}\sin(\varphi_{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})\cdot s_{\mathrm{Q}}(t), \tag{3.44b}$$

where the Anti-Alias Filters (AAF) are considered to be ideal symmetrical low-pass filters with a flat response in the pass-band and an infinite attenuation at the frequency $2\cdot f_{\mathrm{IF}}$.

Inserting (3.43a/b) into (3.44a/b), the received low-pass signal $r(t) = r_{\mathrm{I}}(t) + j\cdot r_{\mathrm{Q}}(t)$ yields

$$r(t) = \frac{1}{\sqrt{\mathrm{N}}}\left(\sum_{k=0}^{\mathrm{N}-1}\left(A_{k,l}\cdot\alpha_1\cdot e^{j\frac{2\pi}{\mathrm{T}}k\left(t-\mathrm{T_G}-l(\mathrm{T}+\mathrm{T_G})\right)} - A_{k,l}^{*}\cdot\alpha_2\cdot e^{-j\frac{2\pi}{\mathrm{T}}k\left(t-\mathrm{T_G}-l(\mathrm{T}+\mathrm{T_G})\right)}\right)\right)\times w\big(t-l\cdot(\mathrm{T}+\mathrm{T_G})\big), \tag{3.45}$$

with

$$\begin{aligned}\alpha_1 = \frac{1}{2}\Bigg\{ &(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})\cdot\left(1+\frac{(1+\gamma_{\mathrm{Q}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})}{(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})}\cos(\varphi_{\mathrm{T}}-\varphi_{\mathrm{R}})\right)+ \\ &+ j\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})\cdot\left(\sin(\varphi_{\mathrm{T}})-\frac{(1+\gamma_{\mathrm{Q}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})}{(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})}\sin(\varphi_{\mathrm{R}})\right)\Bigg\}\end{aligned} \tag{3.46}$$

and

$$\begin{aligned}\alpha_2 = \frac{1}{2}\Bigg\{ &(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})\cdot\left(\frac{(1+\gamma_{\mathrm{Q}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})}{(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})}\cos(\varphi_{\mathrm{T}}-\varphi_{\mathrm{R}})-1\right)+ \\ &+ j\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})\cdot\left(\sin(\varphi_{\mathrm{T}})+\frac{(1+\gamma_{\mathrm{Q}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{I}}^{\mathrm{T}})}{(1+\gamma_{\mathrm{I}}^{\mathrm{R}})\cdot(1+\gamma_{\mathrm{Q}}^{\mathrm{T}})}\sin(\varphi_{\mathrm{R}})\right)\Bigg\}\end{aligned} \tag{3.47}$$

The signal $r(t)$ is sampled as given in (3.6), thus obtaining the samples $r_{n,l}$ which will be fed into the DFT processor, i.e.

$$r_{n,l} = \frac{1}{\sqrt{\mathrm{N}}}\sum_{k=0}^{\mathrm{N}-1}\left(A_{k,l}\cdot\alpha_1\cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot n} - A_{k,l}^{*}\cdot\alpha_2\cdot e^{-j\frac{2\pi}{\mathrm{N}}k\cdot n}\right). \tag{3.48}$$

After DFT, the obtained sequence $Z_{m,l}$ is given by

$$\begin{aligned}Z_{m,l} &= A_{m,l}\cdot\alpha_1 - A_{m,l}^{*}\cdot\alpha_2\cdot\frac{e^{-j\frac{2\pi}{\mathrm{N}}(\mathrm{N}-1)m}}{\mathrm{N}}\cdot\frac{\sin(2\pi m)}{\sin\left(\frac{2\pi}{\mathrm{N}}m\right)} \\ &+\frac{1}{\mathrm{N}}\sum_{\substack{k=0\\(k\neq m)}}^{\mathrm{N}-1} A_{k,l}\cdot\alpha_1\cdot e^{j\frac{\pi}{\mathrm{N}}(k-m)(\mathrm{N}-1)}\cdot\frac{\sin(\pi(k-m))}{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)\right)} \\ &-\frac{1}{\mathrm{N}}\sum_{\substack{k=0\\(k\neq m)}}^{\mathrm{N}-1} A_{k,l}^{*}\cdot\alpha_2\cdot e^{-j\frac{\pi}{\mathrm{N}}(k+m)(\mathrm{N}-1)}\cdot\frac{\sin(\pi(k+m))}{\sin\left(\frac{\pi}{\mathrm{N}}(k+m)\right)},\end{aligned} \tag{3.49}$$

with $0 \le m \le \mathrm{N}-1$. Nevertheless, (3.49) can be simplified to a great extent considering that

$$\frac{\sin(\pi(k+m))}{\sin\left(\frac{\pi}{\mathrm{N}}(k+m)\right)} = \begin{cases}\mathrm{N}; & \text{if } k=\mathrm{N}-m \ (k\neq m)\\ 0; & \text{otherwise}\end{cases}, \tag{3.50}$$

and finally yields

$$Z_{m,l} = \begin{cases} A_{m,l}\cdot\alpha_1 & ;\, m = 0, \frac{\mathrm{N}}{2} \\ A_{m,l}\cdot\alpha_1 + A_{\mathrm{N}-m,l}^{*}\cdot\alpha_2 & ;\,\text{otherwise}\end{cases}. \tag{3.51}$$

The particular form of (3.51), where a sub-carrier m is only affected by the contents in sub-carrier N – m, produces a very characteristic pattern in the constellation diagram when no noise is present, as shown in Figure 3.9. There, the following values are considered: $\gamma_I^T = 0.08$, $\gamma_Q^T = -0.08$, $\gamma_I^R = -0.08$, $\gamma_Q^R = 0.08$, $\varphi_T = 4^o$, $\varphi_R = -4^o$.

In a general case, non-identical anti-alias filters may be modelled by specifying the parameters α_1 and α_2 as a function of the sub-carrier index m, i.e. $\alpha_1(m)$ and $\alpha_2(m)$. However, in order to assure perfect symmetry between the I and Q paths, a digital I/Q generation may be used [Coers01]. Figure 3.10 depicts a possible digital I/Q generation scheme with low IF to be used in the 802.11a and HL2 standards. Under an implementation point of view, the biggest challenge in this alternative solution lies in the design of the Channel Filter in Figure 3.1. This filter requires a very sharp transition band, which is generally achieved by using SAW (Surface Acoustic Wave) filters. Traditionally, IF frequencies in the order of 800 MHz have been used since SAW filters are commercially available at this frequency and not at 20 MHz.

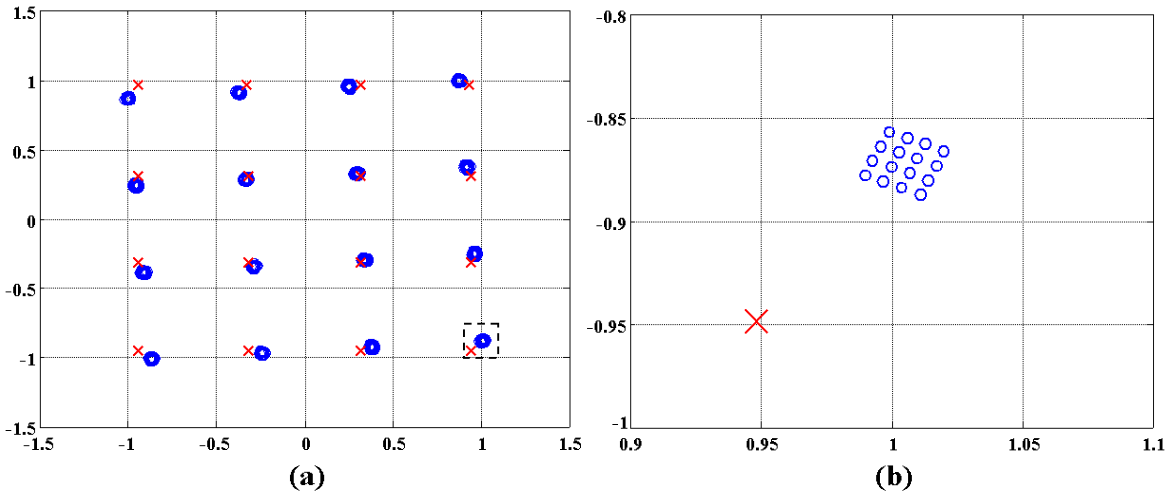


Figure 3.9. a) Effects of an I/Q imbalance on the constellation diagram; b) Zoomed region showing the characteristic pattern.

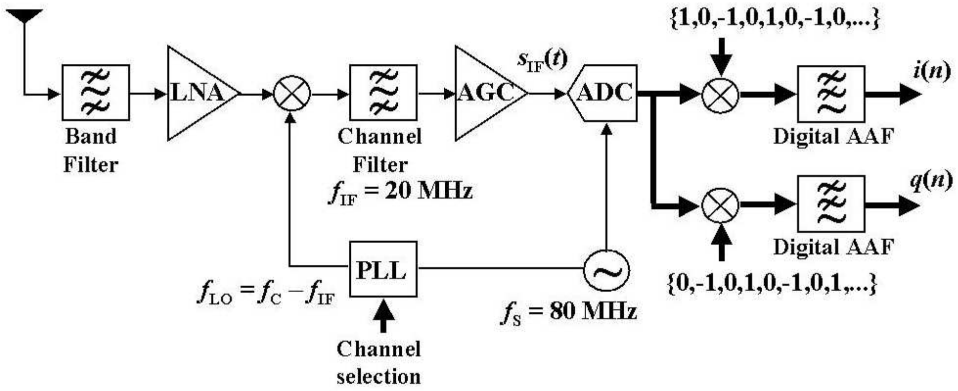


Figure 3.10. Digital I/Q generation valid for the 802.11a and HL2 standards.

According to (3.51), the ICI power derived from an I/Q mismatch is then directly given by $|\alpha_2|^2$, when the data transmitted onto different sub-carriers are uncorrelated and all sub-carriers have the same power σ_A^2. As an example, simulations show that in order to achieve $20\cdot\log_{10}(|\alpha_2|) < -30$dB, the amplitude errors (γ_I^T, γ_Q^T, γ_I^R, γ_Q^R) should be in the range ±0.05 and the phase errors (φ_T, φ_R) in the range ±0.5°.

3.7 Non-linear Distortions

3.7.1 Statistical Behavior of OFDM Signals

OFDM signals can be very well approximated by a complex Gaussian process, since they are the result of an addition of N (or N_u) random variables (Central Limit Theorem). The more sub-carriers are used, the better this approximation will be. Theoretically, a Gaussian random variable can take any value in the interval $(-\infty, \infty)$. However, in a practical implementation this feature is not desirable since real devices have a finite dynamic range. ADCs, mixers and power amplifiers are usually the major sources of non-linear distortions. These distortions can be separated into two groups as suggested in [Ban00]: Cartesian distortions, which act separately on the baseband components of the complex signal, and the AM/AM and AM/PM distortions, which depend on the envelope of the complex signal. The Cartesian distortions are typically due to the ADCs in form of a clipping (saturation) effect. The AM/AM and AM/PM are the amplitude and phase distortions, respectively, which depend on the envelope of the input signal. These distortions are produced by power amplifiers working in their non-linear regions. All these distortions induce ICI components into the OFDM symbols as well as out-of-band radiation, also referred to as *spectral regrowth*.

At this point, it is useful to define the Peak-to-Average Power Ratio (PAPR) of an OFDM symbol as

$$\mathrm{PAPR} \equiv \frac{\max|s_l(t)|^2}{E\{|s_l(t)|^2\}}, \tag{3.52}$$

where $s_l(t)$ is the OFDM symbol l after the DACs, defined similarly as in (3.1) considering a rectangular window without any prefix insertion

$$s_l(t) = \frac{1}{\sqrt{\mathrm{N}}}\sum_{k=0}^{\mathrm{N}-1} A_{k,l}\cdot e^{j2\pi\frac{k}{\mathrm{T}}t}; \qquad 0 \le t \le \mathrm{T}. \tag{3.53}$$

From the definition given in (3.52), it results that the PAPR is a random variable. For further analysis, the Complementary Cumulative Distribution Function (CCDF) of the PAPR is considered [Cim00]. This function is defined as

$$\mathrm{CCDF} = Pr\{\mathrm{PAPR} > \mathrm{PAPR}_0\}. \tag{3.54}$$

In Figure 3.11 and Figure 3.12, the histogram and the CCDF have been represented for the special case of the IEEE 802.11a standard, considering all the possible modulation schemes. In the calculation, the *Preamble* as well as the *Signal* symbols are not considered. The number of generated OFDM symbols was 100,000, each being up-sampled by a factor of four in order to increase the resolution in the calculation of the maximum value.

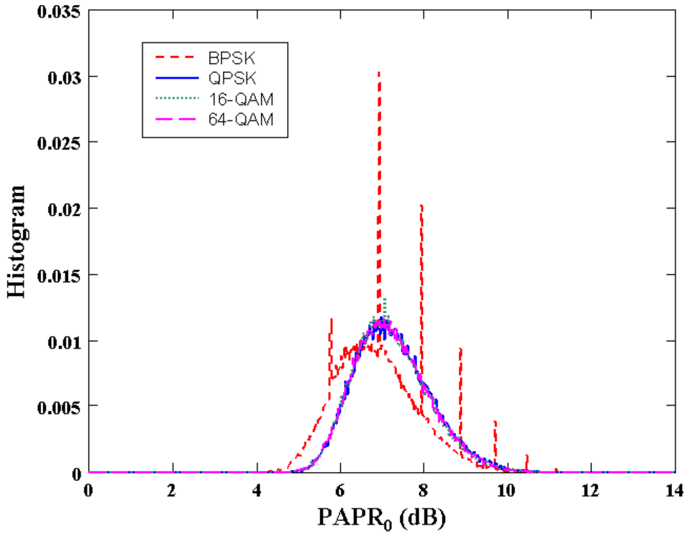


Figure 3.11. Histogram of the OFDM symbols using different modulation schemes in the 802.11a standard.

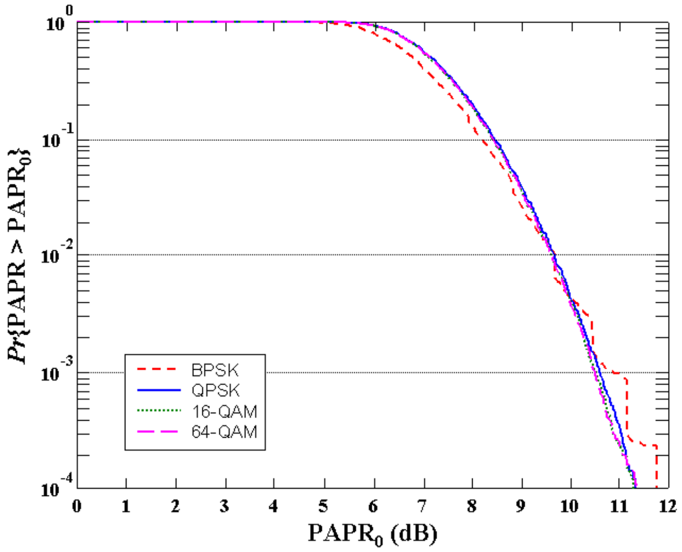


Figure 3.12. CCDF for the different modulation schemes in the 802.11a standard.

In Figure 3.11 it can be seen that in the case of the BPSK modulation scheme, the fact that the Quadrature component is permanently set to 0 makes the histogram of the PAPR to show some peaks.

The CCDF in Figure 3.12 can be used to decide the *input backoff* (IBO) of the power amplifiers. The IBO is the range of allowed input powers for which the amplifier must have a linear response. The worst case will happen when the 6 Mbps transmission mode is being used. In this case, the frame has a maximum length of 1,367 symbols (see Table 2.3). We may impose the restriction that, on average, only one symbol of every 1,367 can be affected by the non-linearities of the amplifiers, which turns into a probability of 1/1,367 = $7.3 \cdot 10^{-4}$. From Figure 3.12, the condition $Pr\{\text{PAPR} > \text{IBO}\} = 7.3 \cdot 10^{-4}$ is satisfied for IBO ≈ 11 dB. However, such a value for the IBO is not achievable without severely decreasing the efficiency of the amplifier, meaning that some kind of pre-distortion algorithm should be used in order to decrease the PAPR of the OFDM symbols. Typical linear amplifier designs cannot achieve values of IBO larger than 4 - 6 dB.

3.7.2 Amplifier Model and Out-of-Band Radiation

The analysis of the ICI and out-of-band radiation is not straightforward and strongly depends on the model used for the High Power Amplifier (HPA). In general, the low-pass equivalent signal at the output of the HPA can be written as

$$c(t) = s(t) \cdot \mathrm{G}(|s(t)|), \tag{3.55}$$

with

$$\mathrm{G}(|s(t)|) \equiv \frac{\mathrm{A}(|s(t)|) \cdot e^{j\Phi(|s(t)|)}}{|s(t)|}. \tag{3.56}$$

The functions A(.) and Φ(.) represent the AM/AM and AM/PM distortions, respectively, and $s(t)$ is the low-pass OFDM signal, which can be written as

$$s(t) = s_{\mathrm{I}}(t) + j \cdot s_{\mathrm{Q}}(t) = r(t) \cdot e^{j \cdot \theta(t)}, \tag{3.57}$$

where the $s_{\mathrm{I}}(t)$ and $s_{\mathrm{Q}}(t)$ components are as given in (3.43a/b). These components are considered to be equal distributed and independent zero-mean stationary Gaussian processes with variance σ^2. The signal $s(t)$ has a variance of $2\sigma^2$. Nevertheless, the amplifier depends on the envelope of $s(t)$, i.e. $|s(t)|$, which is a Rayleigh distributed process with a variance given by [Pro95],

$$\sigma_{\mathrm{R}}^2 = \left(2 - \frac{\pi}{2}\right) \cdot \sigma^2, \tag{3.58}$$

with

$$\sigma^2 = (1 - \alpha_{\mathrm{r}}) \cdot \frac{\mathrm{N_u}}{\mathrm{N}}, \tag{3.59}$$

where N is the number of sub-carriers defined in the OFDM signal, $\mathrm{N_u}$ is the number of “used” sub-carriers (the ones which carry information), and α_{r} is the roll-off factor (α_{r}=0 for the rectangular window). The expression in (3.59) is only valid if the information transmitted in the different sub-carriers inside the symbol, $A_{k,l}$, is uncorrelated and with normalized power ($\sigma_{\mathrm{A}}^2 = 1$).

Considering (3.58), in our analysis the IBO is defined as

$$\mathrm{IBO\,(dB)} \equiv 20 \log_{10}\left(\frac{\mathrm{A_{1dB}}}{\sigma_{\mathrm{R}}}\right), \tag{3.60}$$

where $\mathrm{A_{1dB}}$ is the amplifier’s input amplitude for which the corresponding output is compressed 1 dB with respect to the ideal value.

The amplifier model considered here is the one proposed in [Nee00] and references therein for the Solid-State Power Amplifier (SSPA), i.e.

$$\mathrm{A}(r) = \frac{r}{\left(1 + r^{2p}\right)^{\frac{1}{2p}}} \quad ; \quad \Phi(r) = 0, \tag{3.61}$$

where r represents the envelope of the input signal, and the parameter p is typically in the range of 2 to 3. For the case of p=2, the 1-dB compression point is $A_{1dB} \approx 0.875$. For p=3, this is $A_{1dB} \approx 1.0$. In Figure 3.13, the input-output response of the SSPA with p=2 and p=3 is shown.

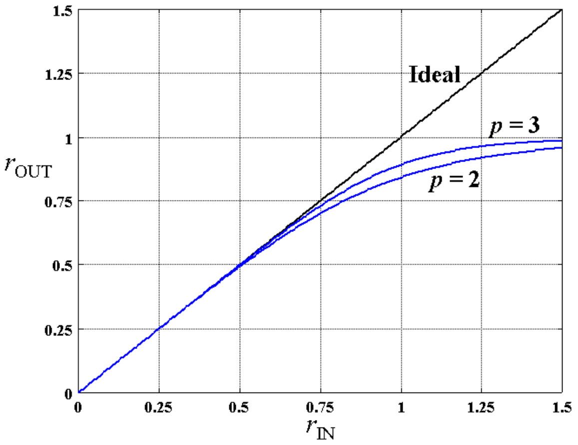


Figure 3.13. Input-output amplitude response of the Solid State Power Amplifier model in (3.61).

In Figure 3.14, simulation results for the Power Spectral Density in the case of the 802.11a and HL2 standards are shown considering the SSPA in (3.61), with p=2 and two values of IBO. In order to adapt the IBO to a specific value, the OFDM symbols $s(t)$ should be multiplied by a parameter G_{IBO} given by

$$G_{IBO} = \frac{A_{1dB}}{\sqrt{\left(2-\frac{\pi}{2}\right)\cdot\sigma^2}}\cdot 10^{-\frac{IBO}{20}}, \tag{3.62}$$

where the IBO is expressed in dB. As expected, when the IBO is 11 dB, no significant spectral regrowth can be appreciated, and the spectrum is still under the limits imposed by the Transmit Spectrum Mask (TSM) [IEEE]. On the contrary, an IBO of 4 dB produces significant out-of-band radiation (according to the TSM) due to the non-linearities.

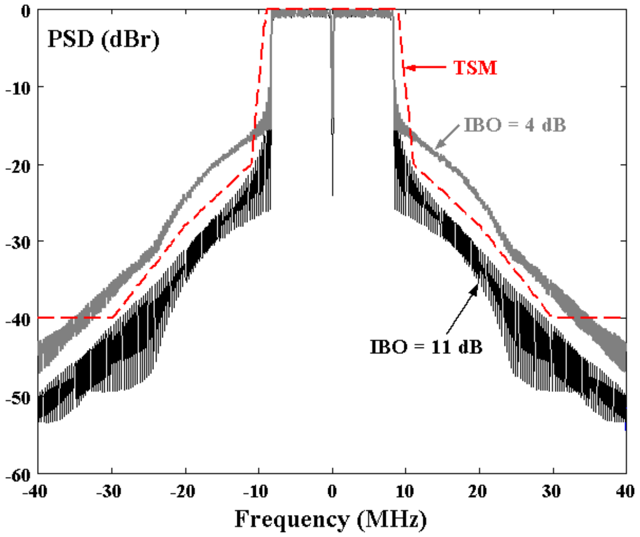


Figure 3.14. Normalized Power Spectral Density after amplification through an SSPA (p=2).

The analysis of the ICI derived from the non-linearities in $c(t)$ (3.55) requires the calculation of the autocorrelation function of $c(t)$. This derivation will not be treated in this Dissertation and the interested reader is referenced to [Ban00], [Costa99] to know more about the mathematical details. The main idea lies on the fact that the signal $c(t)$ in (3.55) may be expressed as

$$c(t) = s(t) + c_d(t), \tag{3.63}$$

where $c_d(t)$ is the distorting signal with power spectral density $\mathrm{C}_d(f)$. Since the PSD of $s(t)$ is known beforehand, $\mathrm{C}_d(f)$ can be easily extracted once the PSD of $c(t)$ has been obtained (Fourier Transform of its autocorrelation function). The ICI power affecting a particular sub-carrier f_k is then approximately given by

$$\mathrm{P}_d(f_k) \approx \mathrm{C}_d(f_k) \cdot \Delta f, \tag{3.64}$$

where Δf stands for the sub-carrier spacing.

3.7.3 Methods to Reduce the PAPR

Several methods have been proposed in the literature in order to reduce the PAPR. These methods can be coarsely classified into the following three groups:

1. Methods based on the multiplication of the transmitted sequence by an optimized code *C* from a set of M possible codes. The multiplication can be realized either before the IDFT, i.e. Selecting Mapping (SLM), or after IDFT, i.e. Partial Transmit Sequence (PTS) [Cim00], [Tar00]. The main advantage of this method is that the bandwidth of the OFDM signal does not increase, since the code *C* is a sequence of phasors with normalized magnitude. However, the major drawback of these methods is the computational complexity they require.
2. Methods based on windowing or “hiding” those peaks exceeding a certain threshold after IDFT [Hen00]. Although simple, the method tends to increase the bandwidth of the OFDM signal. Furthermore, in order to reliably detect the peaks, the OFDM signal has to be up-sampled by at least a factor of 4.
3. Direct compensation of the nonlinear distortions by using an adaptive predistorter [Kang99]. Apart from its hardware complexity, since this method is adaptive in nature, it requires a big number of symbols to be transmitted until the algorithm converges.

The PTS method mentioned above has been applied to the 802.11a standard in order to test its possibilities. Figure 3.15 depicts the system used for this purpose as given in [Cim00]. The mapped sequence $A_{k,l}$ is divided into M disjoint clusters $X_{i,l}$ given by

$$X_{i,l} = \{A_{i\frac{\mathrm{N}}{\mathrm{M}},l}, \ldots, A_{(i+1)\frac{\mathrm{N}}{\mathrm{M}}-1,l}\}, \tag{3.65}$$

with $0 \le i \le \mathrm{M} - 1$. Each cluster contains N values, but only M of them are different of zero. The objective of the PTS approach is to form a weighted combination of the M clusters,

$$X = \sum_{i=0}^{\mathrm{M}-1} b_i X_{i,l}, \tag{3.66}$$

where b_i are the weighting factors and are assumed to be pure rotations. After transforming to the time domain, (3.66) becomes

$$x = \sum_{i=0}^{M-1} b_i x_{i,l} \,. \tag{3.67}$$

The vector $x_{i,l}$, called the *partial transmit sequence*, is the IFFT of $X_{i,l}$. Generally, the weights are selected from the set $\{\pm 1, \pm j\}$ and all the 4^M possible combinations should be tested to find out which of those minimize the PAPR of x. In [Cim00], a suboptimal iterative algorithm is described in which only binary factors $\{\pm 1\}$ are considered and M iterations are necessary. As a first step, it is assumed that $b_i = 1$ for all i. Next, the first phase factor is inverted ($b_0 = -1$) and the PAPR is recomputed. If the new PAPR is lower than in the previous step, the value b_0 is retained as part of the final phase sequence; otherwise, b_0 reverts to its previous value. The algorithm continues in this way until all M possibilities for "flipping" the signs of the phase factors have been explored.

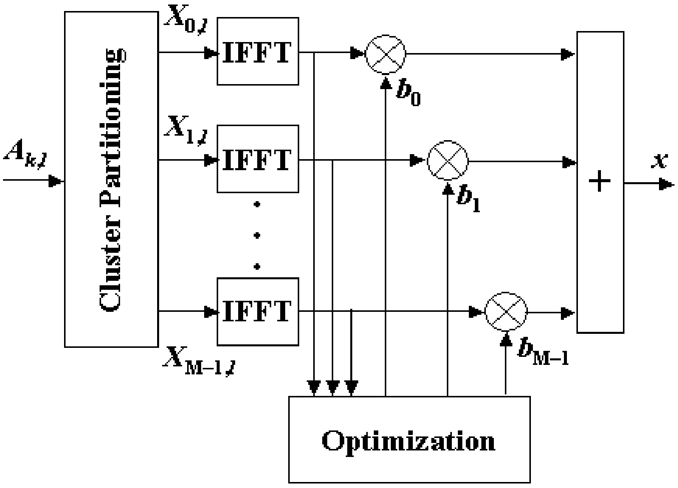
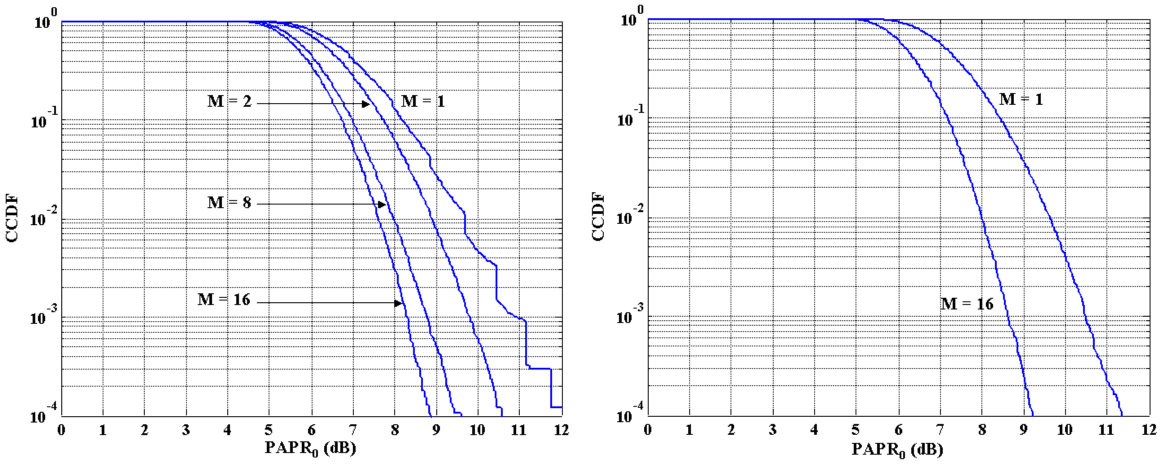


Figure 3.15. Block diagram of the PTS method for PAPR reduction (according to [Cim00]).

Results are shown in Figure 3.16 for the 802.11a standard. Even with M=16 the improvement in the PAPR is only 3 dB for CCDF = 0.01% in the 6 Mbps. In the 54 Mbps, the improvement is only around 2 dB.



(a) (b)

Figure 3.16. CCDF obtained from applying the PTS method in [Cim00] to the 802.11a standard: a) 6 Mbps case; b) 54 Mbps case.

3.8 Summary

In the present Chapter the main impairments affecting an OFDM signal have been investigated in the following order: carrier frequency offset, timing offset, sampling clock frequency offset, phase noise, I/Q mismatches and non-linear distortions.

Regarding the carrier frequency offset, the signal after FFT is affected by both an attenuation-rotation factor and an added ICI component derived from the loss of orthogonality. The rotation factor is constant inside the OFDM symbol but changes from symbol to symbol in a linear way. It is found that the error in the frequency estimation should be less than 1% in order to assure an ISR of about –40dB.

A false determination of the starting point in the OFDM symbols may also lead to considerable penalty. Hence, if the timing error is such that all the considered samples belong to the same OFDM symbol, only a linear phase will appear after calculation of FFT. This is expressed by the condition $N_m - N_G \leq \theta \leq 0$.

The worst case happens when $\theta > 0$. Under this scenario not only the linear phase will be present, but also an ICI component derived from the loss of orthogonality will be added, since not all N samples fed into the FFT belong to the symbol of interest. In addition, the channel coefficient affecting each sub-carrier will be also distorted, leading to erroneous channel estimation.

In sampled systems, a random timing error with an uncertainty in the range $|\theta| < 0.5$ is always present. According to simulation results, a value for $\theta = 0.5$ yields an ISR of about –20 dB. Hence, the interference-induced noise may even become more dominant than the channel noise itself.

A sampling clock frequency error is seen to introduce a variable timing offset. The variation is rather small and becomes noticeable after a large number of OFDM symbols have been received. Since the timing is changing from one symbol to the next one, so it is the slope of the linear phase appearing after FFT. An interesting approach to estimate the clock frequency error based on the slope variation of the linear phase will be given in Chapter 7.

Phase noise has two effects: ICI and CPE. The ICI is again a result of the loss of orthogonality and becomes more important as the number of sub-carriers N increases. The CPE dominates in OFDM systems with a reduced number of sub-carriers and consists of a constant random phase inside the OFDM symbol. This random phase changes randomly on a symbol basis. The correlation between the CPE in two adjacent symbols also increases as N decreases and may be exploited in a CPE estimation algorithm.

Mismatches in the I/Q paths have been analyzed in a simplified manner. Instead of considering non-symmetric Anti-Alias Filters, our approach defines four parameters (γ_I^T, γ_Q^T, γ_I^R, γ_Q^R) to describe amplitude mismatches and two parameters (φ_T, φ_R) to describe phase mismatches. As a result of the I/Q asymmetry it is found that the constellation diagram gets rotated and an ICI component is added. Since a sub-carrier *m* is only affected by data contained in sub-carrier N–*m*, the ICI shows a particular pattern in the absence of AWGN. Simulations show that in order to achieve an ISR of about –30dB in an OFDM system as defined in the 802.11a standard, the amplitude errors (γ_I^T, γ_Q^T, γ_I^R, γ_Q^R) should be in the range ±0.05 and the phase errors (φ_T, φ_R) in the range ±0.5º.

OFDM signals are the result of an addition of N sub-carriers, each modulated by a random process. Hence, by applying the Central Limit Theorem, OFDM signals statistically approach a complex Gaussian random process with an envelope that may take (theoretically) any value from 0 to +∞. The parameter defining this feature is the Peak-to-Average Power Ration (PAPR). As an example, Continuous Wave (CW) signals represent the ideal case since they have a constant envelope, i.e. PAPR = 0 dB. Large PAPRs have a severe impact in actual devices, which show a limited range of linear operation. Non-linear distortions lead to spectral regrowth and generate ICI.

There are two possible ways to tackle the problem. On one hand, one may increase the linear range of the devices, which normally results in power-inefficient systems, e.g. class-A power amplifiers show the best linear performance, but with a power efficiency of only 10% to 30%. Other amplifier classes may be used along with some kind of pre-distortion mechanism. On the other side, one may modify the OFDM system itself to reduce the final PAPR. An example for this is the PTS method, which in the best case shows a PAPR reduction of 3 dB when applied to a system based on the IEEE 802.11a standard.

In the specific case of WLAN systems, where transmission does not take place continuously, class-A or class-AB amplifiers are generally used, regardless of their low power efficiency.

Chapter 4

Synchronization of OFDM Signals

4.1 Introduction

THE RESULTS OBTAINED IN THE FOREGOING CHAPTER SHOW that OFDM signals are very much sensitive to synchronization impairments. The procedures used to eliminate the impairments from the signal take place in the so-called *Inner Receiver*. This block can be divided into two main sub-blocks namely, the *Synchronizer* and the *Channel Estimator*. In this Chapter only the Synchronizer is the subject of our study. Synchronization algorithms suitable for OFDM signals are systematically analyzed and a solution for the specific case of the IEEE 802.11a standard is proposed.

The analysis followed here is based on the division introduced in [Meyr97] for synchronization algorithms, which are clustered into two groups: Decision-Directed (DD) or Data-Aided (DA), and Non-Data-Aided (NDA). Generally, synchronization algorithms depend on the signal being transmitted. If the transmitted data is known at the receiver or it has been previously estimated by some means, the synchronizer is said to be Data-Aided or Decision-Directed, respectively. On the other side, if the transmitted data sequence is not previously known at the receiver, the synchronizer is said to be Non-Data-Aided (NDA).

Considering the various impairments presented in Chapter 3, several questions come to our mind: which of those parameters can be estimated in the Synchronizer?; which is the range of estimation for those parameters?; in which order should this estimation be performed?. In order to answer those questions a thorough analysis of the problem is mandatory. As it will be seen, the problem of the synchronization of OFDM systems started to be treated in a more detailed form at the beginning of the 1990s, when the OFDM transmission technique was being considered for its use in Digital Audio Broadcasting (DAB) systems [DAB]. During the last decade a number of contributions in this field have appeared whenever new OFDM-based systems where envisioned. At the latest stage, OFDM is used in wireless LAN systems aiming to transmit up to 54 Mbps using coherent modulation. This imposes tough demands on the synchronization stage, not only on the precision of the parameter estimation, but also on the allowed power consumption and acquisition speed, since these systems are directed to laptop or handheld portable devices. Therefore it is not only the algorithm itself but also its implementation what has to be carefully considered.

Strictly speaking, the Synchronizer will be responsible for the estimation and correction of the symbol timing (θ) and the carrier frequency offset (ε). A frame detection procedure is also embedded inside the symbol timing estimation when a burst transmission takes place. The results obtained in Chapter 3 show that the symbol timing is in general a variable parameter, mainly due to the sampling clock frequency offset. Similarly, the carrier frequency offset also varies due to the Doppler. Thus, when estimating a parameter it is generally taken the approach in [Meyr97], where each estimation is realized in two stages, i.e. firstly a coarse estimation of the parameter is obtained (acquisition stage) and then it is further refined in a second stage by a fine estimation (tracking stage).

Our analysis of synchronization algorithms for OFDM starts with the NDA methods in §4.2. These methods are generally based on Maximum Likelihood (ML) estimators, which are commonly used due to their favorable statistical properties and because very often they are the only realizable estimators [Kay93]. Nevertheless, the ML estimators show two main disadvantages: limited estimation range and long acquisition time, making them not suitable for WLAN systems. In order to solve these problems one may introduce some side information into the OFDM symbols. Hence, DA synchronization algorithms are possible which will exploit this side information. A thorough analysis of some possible DA mechanisms is realized throughout §4.3. Important results will be derived there for their further application into WLAN systems. Finally, a summary of this Chapter is provided in §4.4.

4.2 Non-Data-Aided Synchronization Algorithms for OFDM Signals

4.2.1 Classical ML Approach

The *Maximum Likelihood* (ML) estimation of a parameter vector $\boldsymbol{\theta} = (\varepsilon,\theta)$ is the value of the vector maximizing the likelihood function $L(\mathbf{r};\boldsymbol{\theta})$. The obtained estimators have the asymptotic properties of being unbiased, achieving the Cramer-Rao Lower Bound (CRLB) and having a Gaussian pdf ([Kay93], page 157).

A first approach for the synchronization of OFDM signals based on the ML method was given by Daffara in [Daf93], where a solution for the frequency offset estimation was proposed. In the recent time, extensions to this approach have been suggested in [Mar01], where also a solution for the timing estimation is proposed by the author and the ML approach is analyzed considering different strategies.

Here we recall the signal model given in (3.1) for an OFDM frame, with the symbols being conformed by a pulse $g(t)$,

$$s(t) = \frac{1}{\sqrt{\mathrm{N}}} \sum_{l=-\infty}^{\infty} \left(\sum_{k=0}^{\mathrm{N}-1} A_{k,l} \cdot e^{j2\pi \frac{k}{\mathrm{T}}\left(t - \mathrm{T_G} - l(\mathrm{T}+\mathrm{T_G})\right)} \right) \cdot g\left(t - l \cdot (\mathrm{T} + \mathrm{T_G})\right). \tag{4.1}$$

For convenience, no distinction between data-bearing and virtual sub-carriers is done in (4.1).

If the frame is transmitted through an AWGN channel, the received signal can be expressed as,

$$r(t) = s((t-t_\theta))e^{j2\pi\frac{\varepsilon}{\mathrm{T}}t} + v(t)\,. \qquad (4.2)$$

where $t_\theta = \theta\cdot(\mathrm{T/N})$ is the timing offset, ε / T is the carrier frequency offset and $v(t)$ stands for zero-mean white Gaussian noise.

The signal in (4.2) is sampled at a rate $1/\mathrm{T_S}$, thus obtaining the samples at the following time instants

$$m\mathrm{T_S} = (n + \mathrm{N_G} + l'\cdot(\mathrm{N} + \mathrm{N_G}))\cdot\frac{\mathrm{T}}{\mathrm{N}}\,, \qquad (4.3)$$

with $l' \in (-\infty, +\infty)$ and $0 \le n \le \mathrm{N}-1$. The sampling interval is selected to be the minimum necessary in order to avoid aliasing, i.e. $\mathrm{T_S} = \mathrm{T/N}$. In the following analysis an observation interval of L=1 symbol (N samples) is considered. The expression in (4.2) now results as

$$r(m\mathrm{T_S}) = s((m-\theta)\mathrm{T_S})e^{j\frac{2\pi}{\mathrm{T}}\varepsilon\cdot m\mathrm{T_S}} + v(m\mathrm{T_S})\,, \qquad (4.4)$$

with

$$s((m-\theta)\mathrm{T_S})e^{j\frac{2\pi}{\mathrm{T}}\varepsilon\cdot m\mathrm{T_S}} = \frac{1}{\sqrt{\mathrm{N}}}e^{j\frac{2\pi}{\mathrm{N}}\varepsilon\cdot(n+\mathrm{N_G})}\left(\sum_{k=0}^{\mathrm{N}-1} A_k \cdot e^{j2\pi\frac{k}{\mathrm{N}}(n-\theta)}\right)\cdot g\left((n-\theta+\mathrm{N_G})\frac{\mathrm{T}}{\mathrm{N}}\right) \qquad (4.5)$$

and $0 \le n \le \mathrm{N}-1$.

The derivation of the ML estimator for the parameter vector $\boldsymbol{\theta} = (\varepsilon, \theta)$ based on the signal model in (4.4) and (4.5) is carried out in Appendix F. The method tries to determine the corresponding frequency and time discriminators e_ε and e_θ, respectively. The resulting scheme is shown in Figure 4.1. As it can be seen, the knowledge of the derivative of the pulse $g(n)$ is required therein. Therefore, this method is not the appropriate one to be used when the conforming pulse $g(n)$ is the rectangular window.

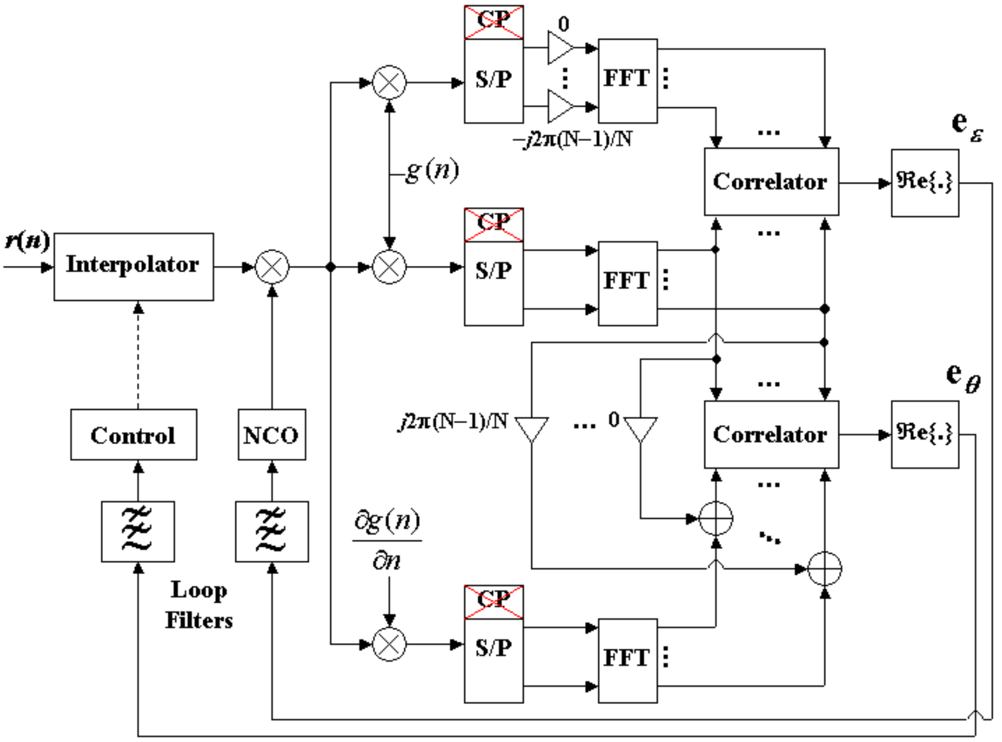
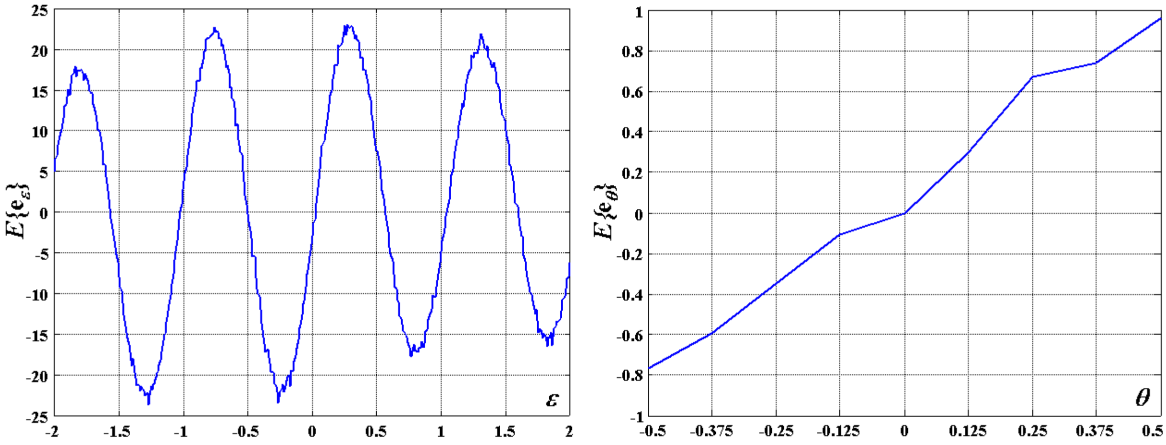


Figure 4.1. General scheme of the classical ML synchronizer.

The s-curves for the discriminators e_ε and e_θ in Figure 4.1 are obtained as $E\{e_\varepsilon\}$ $E\{e_\theta\}$ and depicted in Figure 4.2a/b, respectively. In the calculation, 2,900 symbols were averaged considering a 16-QAM modulation scheme, with $N = 64$, $N_u = 52$, SNR = 10dB and roll-off factor $\alpha_r = 0.25$. In Figure 4.2a, a clear disadvantage of the ML solution can be seen in the fact that the estimation range is limited to $\varepsilon \approx \pm 0.5$ (normalized). The authors in [Daf93] propose an alternative solution in order to increase this frequency estimation range, but at the cost of much higher complexity. Similar results are obtained for the normalized timing estimation θ in Figure 4.2b. However, this limitation is natural since a sampled system cannot estimate timing errors higher than one half its sampling interval. Finally, we should remark that although of discouraging complexity, the scheme in Figure 4.1 is only valid for the AWGN scenario. A simpler method can be obtained if the special statistical properties present in OFDM symbols with cyclic prefix extension are exploited in the definition of the ML solution. This idea was initially given in [Daf95] and is analyzed in the next section.



(a) **(b)**

Figure 4.2. S-curves obtained from the classical ML approach for: a) Frequency discriminator; b) Time discriminator.

4.2.2 Modified ML Solutions

The addition of the cyclic prefix introduces some correlation properties inside the OFDM symbol, which can be exploited by a synchronizer in order to estimate timing and frequency offsets, i.e. the tuple (θ, ε). One of the first contributions to this respect is given by Daffara and Adami [Daf95]. The main idea relies on the fact that when multiplying a sample of the cyclic prefix with the conjugate value of a sample placed N samples later, the result should be a real number in the absence of any frequency offset, since both samples are the same. However, this condition will be only satisfied by those samples in the cyclic prefix not affected by the leakage from the previous symbol due to the channel impulse response. This can be mathematically stated remembering (3.10), which is repeated here for convenience

$$r_{n,l}^{\varepsilon} = e^{j\theta_0} \cdot e^{j\frac{2\pi}{\mathrm{T}} \cdot \varepsilon \cdot \left((n + \mathrm{N_G} + l \cdot (\mathrm{N} + \mathrm{N_G})) \cdot \frac{\mathrm{T}}{\mathrm{N}} \right)} \cdot r_{n,l}, \tag{4.6}$$

where θ_0 is an unknown phase offset and $0 \leq n < \mathrm{N}$, $-\infty < l < \infty$. If no ISI occurs, the samples $r_{n,l}$ are given by

$$r_{n,l} = \frac{1}{\sqrt{\mathrm{N}}} \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot H_{k,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot k\cdot n} + v_{n,l}\,, \tag{4.7}$$

where $v_{n,l}$ are samples of a zero-mean white Gaussian random noise process.

The suggested method relies on the following calculation

$$\begin{aligned}
R_\varepsilon(l) &= r^{\varepsilon}_{n,l} \cdot (r^{\varepsilon}_{n-\mathrm{N},l})^* \\
&= e^{j2\pi\cdot\varepsilon} \cdot r_{n,l} \cdot r^*_{n-\mathrm{N},l} \\
&= \frac{e^{j2\pi\cdot\varepsilon}}{\mathrm{N}} \cdot \left(\sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot H_{k,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot k\cdot n} + v_{n,l} \right) \left(\sum_{\substack{k'=-\mathrm{N_u}/2 \\ (k'\neq 0)}}^{\mathrm{N_u}/2} A^*_{k',l} \cdot H^*_{k',l} \cdot e^{-j\frac{2\pi}{\mathrm{N}}\cdot k'\cdot(n-\mathrm{N})} + v^*_{n-\mathrm{N},l} \right) \\
&= \frac{e^{j2\pi\cdot\varepsilon}}{\mathrm{N}} \cdot \left\{ \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq 0)}}^{\mathrm{N_u}/2} \left|A_{k,l}\right|^2 \cdot \left|H_{k,l}\right|^2 + \right. \\
&\quad + \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq 0)}}^{\mathrm{N_u}/2} \sum_{\substack{k'=-\mathrm{N_u}/2 \\ (k'\neq k; k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot H_{k,l} \cdot A^*_{k',l} \cdot H^*_{k',l} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot(k-k')\cdot n} + \\
&\quad \left. + v_{n,l} \left(\sum_{\substack{k'=-\mathrm{N_u}/2 \\ (k'\neq 0)}}^{\mathrm{N_u}/2} A^*_{k',l} \cdot H^*_{k',l} \cdot e^{-j\frac{2\pi}{\mathrm{N}}\cdot k'\cdot(n-\mathrm{N})} \right) + v^*_{n-\mathrm{N},l} \left(\sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot H_{k,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot k\cdot n} \right) + v_{n,l} \cdot v^*_{n-\mathrm{N},l} \right\}.
\end{aligned} \tag{4.8}$$

From (4.8) it can be observed that the first term in the summation corresponds to a real value and it will not contribute to the phase $e^{j2\pi\varepsilon}$, which is the term of interest. The second term is in general complex and it will disturb the estimation of ε. This term only depends on the transmitted data and is therefore labeled as *self-noise* or *pattern*. The last terms in (4.8) are all derived from the AWGN.

By calculating the expectation of (4.8) it is shown that under the assumption of uncorrelated data, the previous estimator is unbiased, i.e.

$$E\{R_\varepsilon(l)\} = \frac{e^{j2\pi\cdot\varepsilon}}{\mathrm{N}} \cdot \sum_{k=0}^{\mathrm{N}-1} E\{\left|A_{k,l}\right|^2\} \cdot E\{\left|H_{k,l}\right|^2\}\,. \tag{4.9}$$

In order to improve the variance of the estimator in (4.8), L samples are averaged. The value of L corresponds to the number of samples in the cyclic prefix not being affected by the ISI. The normalized frequency offset ε may be estimated either by calculating the phase of $R_\varepsilon(l)$ or by obtaining its imaginary part. However, the previous method presents two main drawbacks. On one hand, no timing error should be present when estimating the frequency offset, since the samples taken from the cyclic prefix should be known beforehand. On the other side, again the maximum range of estimation is limited to $|\varepsilon|<0.5$ given that $-\pi \leq 2\pi\varepsilon < \pi$ in the complex exponential.

A more elaborated solution exploiting the cyclic prefix was given by van de Beek et al. in [Beek97]. In their method they apply the ML estimator to the samples contained in an observation interval of length 2N+N$_G$, as shown in Figure 4.3. No dispersive channel is considered in the analysis, but only the effects of complex, additive, white Gaussian noise.

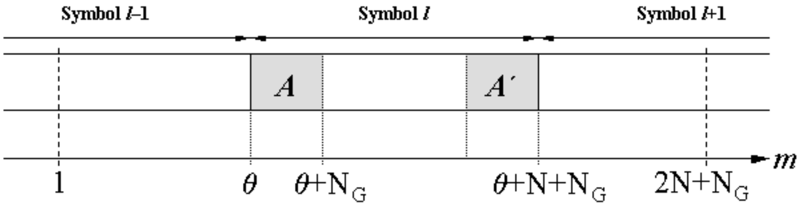


Figure 4.3. Observation interval for the ML estimation problem according to [Beek97].

In order to simplify the notation, a new time variable is defined here as

$$m = n + \mathrm{N_G} + l\cdot(\mathrm{N} + \mathrm{N_G}), \tag{4.10}$$

so as to define the received OFDM signal as

$$r_m = s_{m-\theta} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot\varepsilon\cdot m} + v_m\,, \tag{4.11}$$

s_m being the original transmitted OFDM signal. The signal r_m shows some particular properties on those samples $m\in A\cup A'$. These samples are pairwise correlated as follows,

$$E\{r_m \cdot r_{m+t}^*\} = \begin{cases} \sigma_S^2 + \sigma_N^2 & t = 0 \\ \sigma_S^2 \cdot e^{j2\pi\varepsilon} & t = \mathrm{N} \\ 0 & \text{otherwise} \end{cases}\,, \tag{4.12}$$

for $\forall m\in A$; where $\sigma_S^2 = \gamma E\{|s_m|^2\}$, $\sigma_N^2 = E\{|v_m|^2\}$, and $\gamma = (\mathrm{N}-\mathrm{N_v})/\mathrm{N}$ (N$_v$ is the number of virtual sub-carriers being used in the OFDM system).

As shown in Appendix F, the ML estimator for a vector parameter (θ, ε) is the value maximizing the joint probability density function, i.e.

$$(\hat{\theta},\hat{\varepsilon}) = \arg\max_{\theta,\varepsilon}\{p(\mathbf{r}\,;\theta,\varepsilon)\}\,, \tag{4.13}$$

which is equivalent to (F.5). However, in this case the vector **r** is formed by the 2N+N$_G$ samples found in the observation interval, i.e. $\mathbf{r} = (r_1 \ldots r_{2\mathrm{N+NG}})^{\mathrm{T}}$. In (4.13) it is considered that the unknown parameters (θ, ε) are deterministic. In order to solve the stated problem, a low SNR scenario it is not considered as it was in (F.8). Instead, a new function called the *log-likelihood function* is defined as

$$\Lambda(\theta,\varepsilon) = \log p(\mathbf{r}\,;\theta,\varepsilon)\,. \tag{4.14}$$

The analysis shown in [Beek97] results in the following equation for $\Lambda(\theta, \varepsilon)$,

$$\begin{aligned}\Lambda(\theta,\varepsilon) &= \Re\mathrm{e}\left\{e^{j2\pi\varepsilon}\sum_{m=\theta}^{\theta+\mathrm{N_G}-1} r_m^* \cdot r_{m+\mathrm{N}}\right\} - \frac{\rho}{2}\sum_{m=\theta}^{\theta+\mathrm{N_G}-1}(|\,r_m\,|^2 + |\,r_{m+\mathrm{N}}\,|^2) \\ &= |\,J(\theta)\,|\cos(2\pi\varepsilon + \angle J(\theta)) - \rho E(\theta)\,,\end{aligned} \tag{4.15}$$

with

$$J(\theta) = \sum_{m=\theta}^{\theta+\mathrm{N_G}-1} r_m^* \cdot r_{m+\mathrm{N}} \,, \tag{4.16}$$

$$E(\theta) = \frac{1}{2} \sum_{m=\theta}^{\theta+\mathrm{N_G}-1} (|\, r_m \,|^2 + |\, r_{m+\mathrm{N}} \,|^2) \,, \tag{4.17}$$

$$\rho \equiv \frac{\sigma_S^2}{\sigma_S^2 + \sigma_N^2} = \frac{\mathrm{SNR}}{\mathrm{SNR}+1} \,. \tag{4.18}$$

At this point we focus our interest on those values for the tuple (θ, ε) maximizing the log-likelihood function in (4.15). The maximum value will occur when the cosine function equals 1, i.e.

$$\cos(2\pi\varepsilon + \angle J(\theta)) = 1 \,, \tag{4.19}$$

and the likelihood function yields

$$\Lambda(\theta,\varepsilon) = |\, J(\theta) \,| - \rho E(\theta) \,. \tag{4.20}$$

Hence, the general procedure to obtain the ML estimations for timing and frequency (θ_{ML}, $\varepsilon_{\mathrm{ML}}$) will be as follows: first the value θ_{ML} maximizing the function in (4.20) has to be found. Afterwards, the phase of $J(\theta)$ at $\theta = \theta_{\mathrm{ML}}$ will provide the estimation of the normalized frequency offset $\varepsilon_{\mathrm{ML}}$. From (4.19) this estimation can be written in an exact form as

$$\varepsilon = -\frac{\angle J(\theta)}{2\pi} + z \,, \tag{4.21}$$

where z is an integer number. The value z introduces uncertainty into the estimation and restricts once again the range of possible estimation values to the interval $|\varepsilon| < 0.5$.

A block diagram for this modified ML solution is sketched in Figure 4.4. As it is clear from equation (4.16), the function $J(\theta)$ corresponds to a correlation term, which was already found when presenting the solution in [Daf95]. The solution proposed by Daffara and Adami in [Daf95] suffers from self-noise, i.e. not only the noise will interfere the estimation but also the information carried into the OFDM symbols. In this context, the solution in [Beek97] tries to indirectly assess the instantaneous self-noise through the calculation of $E(\theta)$ in order to improve the estimation of the timing. Nevertheless, $E(\theta)$ is weighted by the parameter ρ, which depends on the Signal-to-Noise Ratio, i.e. when it is found that a low SNR is present in the channel, the synchronizer will not rely much on its own estimation of the instantaneous self-noise. In Figure 4.5, the functions $J(\theta)$, $\Lambda(\theta)$ as well as $\varepsilon_{\mathrm{ML}}(\theta)$ are shown for the case of a noiseless channel with $\varepsilon = 0.2$ in the case of an IEEE 802.11a system with 36 Mbps transmission rate.

The architecture in Figure 4.4 represents a major simplification with respect to the one in Figure 4.1. However, there are several drawbacks associated with this modified ML solution. For instance, the value of ρ is generally unknown at the receiver, meaning that either it has to be estimated at the synchronizer (increase of the computational burden), or it has to be kept to a constant value considering an average SNR in the channel (sub-optimum ML estimation). Nevertheless, the main weakness falls in the fact that in the statement of the problem no channel model has been considered. The authors in [Beek97] propose an averaging of several likelihood functions in order to improve the results when a multipath channel is present, but the results shown there illustrate very poor performance with respect to the timing estimation.

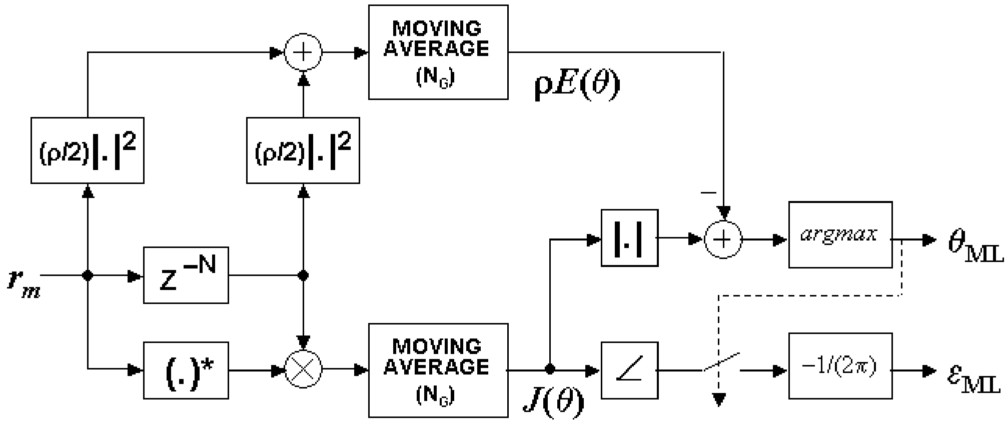


Figure 4.4. Block diagram of the modified ML estimator as proposed by van de Beek [Beek97].

Variations to the previous modified ML scheme have been further proposed in the literature. In [Beek95b] the authors are aware of the computational complexity found in this solution and try to derive a simplified algorithm, which is nevertheless only valid for timing estimation. Instead of taking the full bunch of bits used for representing r_m in (4.11), only the sign is considered. The obtained signal c_m is given by

$$c_m = sign(r_m) = sign(\Re e\{r_m\}) + j \cdot sign(\Im m\{r_m\}) \,, \tag{4.22}$$

where

$$sign(x) = \begin{cases} +1; & x \geq 0 \\ -1; & x < 0 \end{cases} . \tag{4.23}$$

Under these conditions, the log-likelihood function in (4.15) simplifies as follows (valid for timing estimation only),

$$\Lambda_C(\theta) = \sum_{m=\theta}^{\theta+\mathrm{N_G}-1} \Re e\{c_m^* \cdot c_{m+\mathrm{N}}\} \tag{4.24}$$

and

$$\theta_C = \arg\max\{\Lambda_C(\theta)\} \,. \tag{4.25}$$

Although simple, the performance of the metric (4.24) is very poor even in the AWGN channel, as shown in [Beek95b]. This performance improves significantly if several log-likelihood functions are averaged, even under fading channel conditions, but only when no frequency offset is present. In order to make this simple solution also robust to frequency offsets and even use it for the frequency offset estimation, several modifications are proposed in [Hsieh99]. Considering the simplified autocorrelation function

$$J_C(\theta) = \sum_{m=\theta}^{\theta+\mathrm{N_G}-1} c_m^* \cdot c_{m+\mathrm{N}} \,, \tag{4.26}$$

the log-likelihood function is modified as follows in order to make it robust against frequency offsets, i.e.

$$\Lambda_C(\theta) = \left|\Re e\{J_C(\theta)\}\right| + \left|\Im m\{J_C(\theta)\}\right| , \tag{4.27}$$

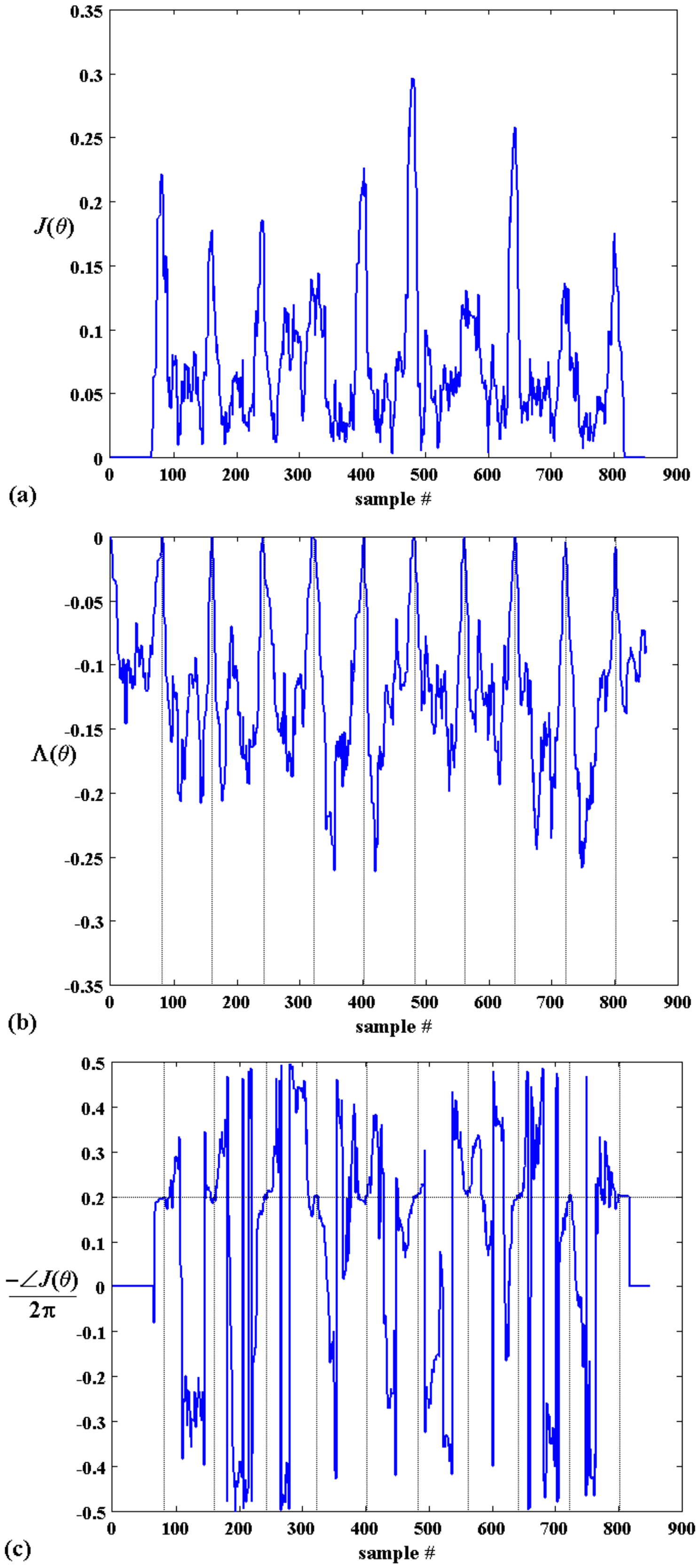


Figure 4.5. Signals generated by the ML estimator (N=64, N_G=16, $\varepsilon = 0.2$, SNR=∞, 16QAM): a) Magnitude of the autocorrelation $J(\theta)$; b) Log-likelihood function $\Lambda(\theta)$; c) Phase of the autocorrelation $J(\theta)$.

so as to obtain the timing estimation θ_C as in (4.25), but using therein the expression in (4.27) for $\Lambda_C(\theta)$. The frequency estimation ε_C is obtained as

$$\varepsilon_C = -\frac{\angle J_C(\theta)}{2\pi} + z\,, \tag{4.28}$$

which is also restricted to the range $|\varepsilon_C| < 0.5$. In [Hsieh99] the estimation range for the frequency offset is greatly extended by considering a data-aided approach, which will be analyzed in more detail in §4.3. A diagram of the solution presented in [Hsieh99] is depicted in Figure 4.6.

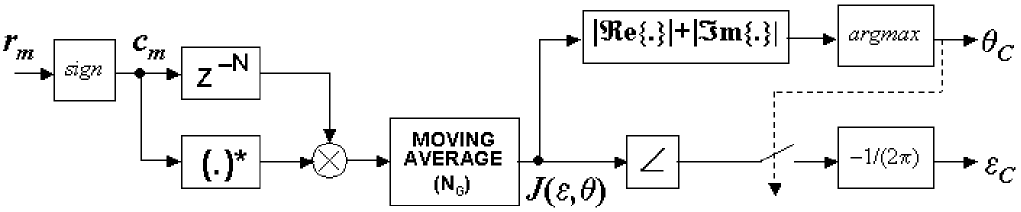


Figure 4.6. Simplified scheme of the modified ML estimator according to [Hsieh99].

The methods seen so far are an attempt to estimate timing and frequency offsets in one single architecture. Other authors have tried to find specific solutions for either the timing or the frequency offset estimations considering that the other parameter is known beforehand. Thus, in [Land99] the authors suggest an ML timing estimator based not only on the redundancy introduced by the cyclic prefix, but also on the existence of pilot information, i.e. it is no longer a pure NDA scheme. The obtained solution achieves a big improvement in the timing estimation, i.e. reduced estimation variance, compared to the solution in [Beek97]. The main reason for the better performance is that the pilot information further helps in the estimation of the self-noise. As a consequence, in order to obtain the same performance, the algorithm in [Land99] requires a shorter cyclic prefix as compared to the one in [Beek97]. The derivation is done considering a signal model as follows,

$$r_m = s_{m-\theta} + p_{m-\theta} + v_m\,, \tag{4.29}$$

where p_m is a known sequence, and no frequency offset is present. The mathematical analysis is exactly the same as the one carried out for the modified ML derivation in [Beek97], but with the difference that the samples r_m now show a statistical mean equal to $p_{m-\theta}$, which has to be considered in the calculation of the joint PDF.

Although the full derivation is done only under assumption of an AWGN channel, the algorithm shows good performance under fading channel conditions and small non-corrected frequency offsets. Nevertheless, the computational burden is more than twice compared with the solution in [Beek97].

4.3 Data-Aided Synchronization Algorithms for OFDM Signals

The results obtained in the foregoing section showed three main negative aspects:

1. The frequency offset estimation range is restricted to be less than half of the channel spacing. Although the ML solution proposed in [Daf93] allows an extension of this acquisition range by filtering the likelihood function, this is only at the expense of increased complexity.
2. Timing estimators based on the cyclic prefix extension are not accurate enough in a multipath scenario. In multipath fading channels, the guard interval is corrupted by the ISI. In this case, the correlation method based on the guard interval will typically result in a delayed estimation of the timing, proportional to the *center of gravity* of the channel, i.e. the root-mean-square value of the time delay (τ_{rms}).
3. Yet another disadvantage is the long acquisition time required by these methods, since they require an average over several estimations.

The only way to overcome these problems is by providing further information to the synchronization process, as it will be seen in the following sub-sections, where improvements in the frequency and timing estimations are proposed.

4.3.1 Extension of the Frequency Estimation Range

In view of the fact that the maximum estimation range is half of the sub-carrier spacing Δf, one may get tempted to *virtually* increase Δf in order to expand this estimation range. This is possible if a number of sub-carriers are forced to be 0 before the calculation of the IDFT in the transmitter. Thus, in order to increase the estimation range by a factor of 2, one may force to 0 the sub-carriers corresponding to $k = 0,2,4,6\ldots$ As a result, the obtained N-sample symbol will show a periodicity of N/2 samples. In general, in order to increase the estimation range by a factor L, a number of sub-carriers equal to $\mathrm{N}\cdot(1-\mathrm{L}^{-1})$ has to be forced to 0, generating an OFDM symbol showing L periods of M samples each, with $\mathrm{N}=\mathrm{M}\cdot\mathrm{L}$. In this manner, the frequency estimation range is extended to $|\varepsilon|<(\mathrm{L}/2)$. A solution exploiting this feature can be found in [Mor99], where the correlation properties of a particular symbol showing several periodicities are exploited. As it was shown in (4.12) for the case of a symbol with a cyclic prefix, now it is possible to obtain the correlation function of a symbol showing several periodicities, when no cyclic prefix is considered (perfect timing is supposed), as follows

$$R(l) = E\{r_m \cdot r_{m+l}^*\} = \begin{cases} \sigma_S^2 + \sigma_V^2 & l = 0 \\ \sigma_S^2 \cdot e^{j\frac{2\pi}{\mathrm{L}} l\cdot\varepsilon} & l = 1, 2, 3, \ldots, (\mathrm{L}-1)\mathrm{M} \\ 0 & \text{otherwise} \end{cases} \quad (4.30)$$

In [Mor99] the following estimator for the correlation function in (4.30) is proposed,

$$\hat{R}(l) = \frac{1}{\mathrm{N} - l\cdot\mathrm{M}} \sum_{m=l\cdot\mathrm{M}}^{\mathrm{N}-1} r_m r_{m-l\cdot\mathrm{M}}^* , \quad (4.31)$$

for $0 \le l \le \mathrm{H}$; H being a design parameter less than or equal to L–1.

The information of interest is in the phase of $R(l)$, since it only depends on ε. Thus, the following sequence may be extracted,

$$\varphi(l) = [\angle\hat{R}(l) - \angle\hat{R}(l-1)]_{2\pi}, \tag{4.32}$$

for $1 \le l \le \mathrm{H}$. The sub-index states for a modulo-2π operation.

Considering a signal model similar to the one in (4.11) with $\theta = 0$, i.e.

$$r_m = e^{j\frac{2\pi}{\mathrm{N}}\varepsilon\cdot m} s_m + v_m, \tag{4.33}$$

the operation described in (4.31) results as

$$\begin{aligned}\hat{R}(l) &= \frac{1}{\mathrm{N}-l\cdot\mathrm{M}}\sum_{m=l\cdot\mathrm{M}}^{\mathrm{N}-1}\left(e^{j\frac{2\pi}{\mathrm{N}}\varepsilon\cdot m}s_m + v_m\right)\left(e^{j\frac{2\pi}{\mathrm{N}}\varepsilon(m-l\cdot\mathrm{M})}s_{m-l\cdot\mathrm{M}} + v_{m-l\cdot\mathrm{M}}\right)^*\\ &= \frac{1}{\mathrm{N}-l\cdot\mathrm{M}}\sum_{m=l\cdot\mathrm{M}}^{\mathrm{N}-1}\left(e^{j\frac{2\pi}{\mathrm{N}}\varepsilon\cdot m}s_m + v_m\right)\left(e^{j\frac{2\pi}{\mathrm{N}}\varepsilon(m-l\cdot\mathrm{M})}s_{m-l\cdot\mathrm{M}} + v_{m-l\cdot\mathrm{M}}\right)^*\\ &= \frac{e^{j\frac{2\pi}{\mathrm{N}}\varepsilon\cdot l\cdot\mathrm{M}}}{\mathrm{N}-l\cdot\mathrm{M}}\sum_{m=l\cdot\mathrm{M}}^{\mathrm{N}-1}\left(|s_m|^2 + s_m\cdot\tilde{v}^*_{m-l\cdot\mathrm{M}} + s^*_{m-l\cdot\mathrm{M}}\cdot\tilde{v}_m + \tilde{v}_m\cdot\tilde{v}^*_{m-l\cdot\mathrm{M}}\right),\end{aligned} \tag{4.34}$$

where

$$\tilde{v}_m = v_m\cdot e^{-j\frac{2\pi}{\mathrm{N}}\varepsilon\cdot m}, \tag{4.35}$$

is statistically equivalent to v_m.

In (4.34) it has been considered that s_m and $s_{m-l\cdot\mathrm{M}}$ are identical, thus eliminating any self-noise component from $\hat{R}(l)$. This can be seen if the product of these two signals is analyzed in more detail as follows,

$$\begin{aligned}&\sum_{m=l\cdot\mathrm{M}}^{\mathrm{N}-1} s_m\cdot s^*_{m-l\cdot\mathrm{M}} =\\ &\quad= \sum_{m=l\cdot\mathrm{M}}^{\mathrm{N}-1}\left(\sum_{k=0}^{\mathrm{N}-1}|A_k|^2\cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot l\cdot\mathrm{M}} + \sum_{k=0}^{\mathrm{N}-1}\sum_{k'\neq k}^{\mathrm{N}-1}A_k A^*_{k'}\cdot e^{j\frac{2\pi}{\mathrm{N}}(k-k')\cdot n}\cdot e^{j\frac{2\pi}{\mathrm{N}}k'\cdot l\cdot\mathrm{M}}\right)\\ &\quad= (\mathrm{N}-l\cdot\mathrm{M})\cdot\sum_{k=0}^{\mathrm{N}-1}|A_k|^2\cdot e^{j\frac{2\pi}{\mathrm{L}}k\cdot l} + \sum_{k=0}^{\mathrm{N}-1}\sum_{k'\neq k}^{\mathrm{N}-1}A_k A^*_{k'}\cdot e^{j\frac{2\pi}{\mathrm{L}}k'\cdot l}\cdot e^{j\frac{\pi}{\mathrm{N}}(k-k')\cdot(l\cdot\mathrm{M}-1)}\,\frac{\sin\left(\frac{\pi}{\mathrm{L}}(k-k')\cdot l\right)}{\sin\left(\frac{\pi}{\mathrm{N}}(k-k')\right)}\end{aligned} \tag{4.36}$$

In (4.36) it has to be noticed that by definition $k = \mathrm{L}\cdot z$ (z being an integer) in order to obtain the periodicities in the OFDM symbol. By this feature, the sine function in the second summand will be 0, eliminating the self-noise. Furthermore, the exponential found in the first summand will always be 1.

The analysis presented in [Mor99] suggests the following linear estimator for the normalized frequency offset,

$$\hat{\varepsilon} = \frac{\mathrm{L}}{2\pi}\sum_{l=1}^{\mathrm{L}/2} w_l \cdot \varphi(l)\,, \tag{4.37}$$

where the weights w_l are given by the following expression,

$$w_l = \frac{12(\mathrm{L}-l)(\mathrm{L}-l+1)-3\mathrm{L}^2}{2\mathrm{L}(\mathrm{L}^2-1)}\,. \tag{4.38}$$

Another way to increase the frequency estimation range is by analyzing the information in the frequency domain. A first contribution on this regard was given by Moose in [Moose94]. In his method two identical symbols of length N, $r_{m,1}$ and $r_{m,2}$, are repeated in time. Under this situation, and considering that no timing error is present, the estimation of the frequency offset can be realized in the frequency domain. Recalling the fact that a frequency offset can be seen as a constant phase inside an OFDM symbol, and that this phase changes linearly on a symbol basis -see result shown in (3.12)-, the frequency information of two identical symbols affected exclusively by a frequency offset is given as follows

$$\begin{aligned} R_{k,1} &= Z_{k,1} + V_{k,1} \\ R_{k,2} &= Z_{k,1}e^{j2\pi\varepsilon} + V_{k,2} \end{aligned} \tag{4.39}$$

for $\forall k \in K$, where K represents the set of data-bearing sub-carriers. The samples $Z_{k,1}$ and $Z_{k,2}$ correspond to the first term in (3.12) , which are the noise-free components affected by the amplitude and phase distortion. The samples $V_{k,1}$, $V_{k,2}$ are zero-mean Gaussian noise components and already include the ICI (second and third terms in (3.12)). Considering the signal model in (4.39), [Moose94] derives the ML estimator for ε as

$$\varepsilon_{\mathrm{ML}} = \frac{1}{2\pi}\tan^{-1}\left(\frac{\Im\mathrm{m}\{\sum_{k\in K} R_{k,2}R_{k,1}^*\}}{\Re\mathrm{e}\{\sum_{k\in K} R_{k,2}R_{k,1}^*\}}\right), \tag{4.40}$$

where no longer appears the self-noise component found in (4.8). Although a better estimator, the expression in (4.40) still suffers from the range limitation, i.e. $|\varepsilon|<0.5$, and is suboptimal in the sense that two symbols need to be transmitted, when other methods seen so far require one single symbol for the same estimation range. Nevertheless, this method provides an interesting idea, which was further exploited in forthcoming papers.

The methods seen up to this point consider the data content in the OFDM symbols to be completely random at the receiver side. As an attempt to improve the synchronization algorithms, a number of authors started to consider alternative schemes in which a number of sub-carriers were reserved to transmit known information. A first contribution in this respect was given in [Class94]. Therein, the authors suggest a method by which some of the transmitted OFDM symbols contain a number of *pilot* sub-carriers, which are spread uniformly over the whole frequency domain. Figure 4.7 provides an example of a possible distribution of the pilots in time and frequency. The scheme considers that 1 of each D-th transmitted symbol contains pilot sub-carriers. The pilot-bearing sub-carriers are always at the same frequency positions and are placed Δ sub-carriers apart from each other. Obviously, the bigger Δ is, the more unreliable the frequency estimation is going to be since fewer pilots will be available.

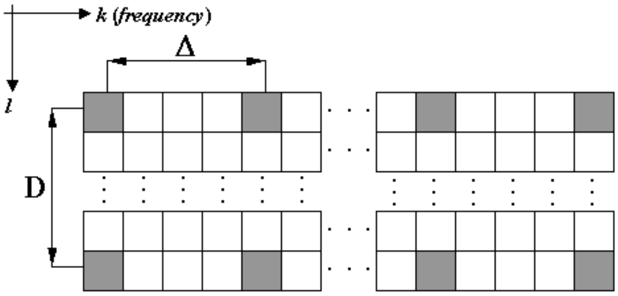


Figure 4.7. Pilot distribution according to [Class94] in order to perform frequency estimations.

Similarly to the estimator in [Moose94], the frequency offset is obtained in this case as,

$$\varepsilon = \frac{1}{2\pi \mathrm{D}} \angle J_{\mathrm{F}}, \tag{4.41}$$

with the correlation function J_{F} defined as,

$$J_{\mathrm{F}} = \sum_{k \in P} (P_{k,l} \cdot C_{k,l}^{*}) \cdot (P_{k,l+\mathrm{D}} \cdot C_{k,l+\mathrm{D}}^{*})^{*} . \tag{4.42}$$

The estimation range can be seen to be $|\varepsilon|<(2\mathrm{D})^{-1}$, so in principle it may appear that this method does not provide much improvement compared to Moose's solution in [Moose94]. In (4.42) P represents the set of possible values for the frequency sample k containing pilots, $C_{k,l}$ is the expected value for a particular pilot on symbol l and sub-carrier k, and $P_{k,l}$ is the actual received value for that pilot. In order to improve the estimation range in (4.41), the authors in [Class94] established for the first time a method based on correlations in the frequency domain. Since the correlation function in (4.42) is maximized when no frequency offset is present, the authors suggest a try-and-change algorithm in which a certain frequency offset (f_{trial}) is applied intentionally and iteratively modified until the magnitude of J_{F} is maximized. Unfortunately, the time required by such frequency estimations may increase significantly if the trial frequency is modified in very small steps.

For applications in which estimation time is critical, i.e. those based on burst transmissions, the only way to achieve a rapid estimation together with an extended estimation range is by using dedicated synchronization symbols in which no payload data are transmitted at all. Let's consider the normalized frequency offset ε to be decomposed into two parts, i.e. $\varepsilon = \alpha + \beta$, being α a fine (or fractional) frequency offset and β a coarse (or integer) frequency offset. If α can be estimated and corrected, the symbols will only show a frequency offset of β. Since β is an integer, the orthogonality inside the OFDM symbols will be preserved and after FFT no ICI but just a frequency shift will be observed. In view of this, Schmidl and Cox [Schmidl97] suggested the use of two uniquely dedicated synchronization symbols. The frequency contents of these *preamble* symbols are two PN (Pseudo-Noise) sequences, $C_{k,1}$ and $C_{k,2}$, with good correlation properties (see [Pro95, p.725] and references therein on how to generate PN sequences). The samples for $C_{k,1}$ and $C_{k,2}$ are taken from a QPSK constellation.

In the frequency domain, the first preamble symbol contains a PN sequence $C_{k,1}$ only on the *even* sub-carriers (excluding DC), while the odd sub-carriers are forced to 0. Thus, this symbol shows a periodicity of N/2 samples in the time domain (see Figure 4.8a). Therefore, the time-correlation method presented in [Mor99] (with L=2) may be applied to estimate α in the range $|\alpha|<1$. Further on, the two preamble symbols have to be corrected for the fractional frequency offset using a Numerically Controlled Oscillator (NCO), hence recovering orthogonality inside the symbols.

After correction for α, the uncertainty in the frequency offset is reduced to $\varepsilon = \beta$, with $\beta = 2z$ (z an integer). In order to estimate β, the FFT of both symbols is obtained as $P_{k,1} = H_{k,1} \cdot C_{k,1} + V_{k,1}$ and $P_{k,2} = H_{k,2} \cdot C_{k,2} + V_{k,2}$ ($V_{k,1}$, $V_{k,2}$ standing for the noise terms), and a crosscorrelation in the frequency domain with the reference sequence $C_{k,\mathrm{REF}} = C_{k,2} \cdot (C_{k,1})^*$ is carried out. The estimation of β will then be calculated as follows,

$$\hat{\beta}_{\mathrm{S\&C}} = \underset{g}{\arg\max} \left| \sum_{k} (P_{k+2g,2} \cdot P^*_{k+2g,1}) \cdot (C_{k,2} \cdot C^*_{k,1})^* \right| . \tag{4.43}$$

Note that in the presence of a multipath fading channel, the estimator in (4.43) will not be biased whenever the channel is approximately constant for the duration of the two preamble symbols, i.e. $H_{k,1} \approx H_{k,2}$. Even if a linear phase derived from a timing offset is present, the estimator in (4.43) will not be affected since this linear phase will be the same on both symbols -see (3.22)- and it will be compensated through the conjugate operation. This method has to be seen as a hybrid solution, which cleverly combines the ideas asserted in [Mor99], [Moose94] and [Class94]. In this manner Schmidl and Cox established a preamble structure, which has been exploited in the most recent definitions of WLAN standards.

An interesting improvement in the way of using two preamble symbols for frequency estimation is provided by Stantchev and Fettweis in [Stant98]. In this particular solution, both preamble symbols contain exactly the same PN sequence $C_{k,\mathrm{P}}$ (here the subscript P stands for "Preamble") and do not show any periodicity, i.e. all sub-carriers except the virtual ones contain a value of the PN sequence. Nevertheless, a cyclic prefix is added to the first preamble symbol as shown in Figure 4.8b. The first symbol is used to extract the fractional frequency offset, which in this case is restricted to the range $|\alpha|<0.5$. The fine frequency offset α is estimated based on the solution shown in Figure 4.4, exploiting the existence of the cyclic prefix. During synchronization, only the second preamble symbol is afterwards corrected for α and converted into the frequency domain through an FFT. The coarse frequency offset β may take now any integer value (not just a multiple of 2 as it happened in Schmidl & Cox's solution). The reference signal used in the crosscorrelator is at this point the sequence $C_{k,\mathrm{REF}} = C_{k+1,\mathrm{P}} \cdot (C_{k,\mathrm{P}})^*$, with $k = 0,\ldots, \mathrm{N}-2$, resulting in the following estimator for β,

$$\hat{\beta}_{\mathrm{S\&F}} = \underset{g}{\arg\max} \left| \sum_{k=0}^{\mathrm{N}-2} (P_{k+1+g,\mathrm{P}} \cdot P^*_{k+g,\mathrm{P}}) \cdot (C_{k+1,\mathrm{P}} \cdot C^*_{k,\mathrm{P}})^* \right| . \tag{4.44}$$

Stantchev's solution is faster than the one in [Schmidl97] since only one FFT needs to be calculated. Furthermore, since a cyclic prefix is considered, no ISI-induced noise will deteriorate the estimations in the presence of fading channels.

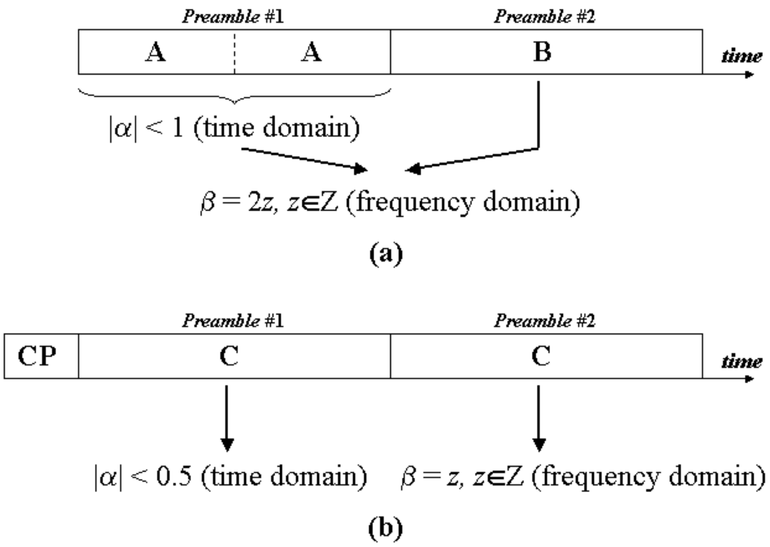


Figure 4.8. Preamble structures for increased frequency estimation range: a) according to Schmidl & Cox [Schmidl97]; b) according to Stantchev [Stant98].

4.3.2 Coarse Timing Estimation

The idea introduced in §4.3.1 where the normalized frequency offset ε was divided into fine (α) and course (β) offsets can also be applied in the case of timing estimation. The timing estimation problem is therefore divided into three operations, namely *coarse timing estimation*, *fine timing estimation* and *tracking*. The coarse timing estimation will be discussed in this section.

Under coarse timing estimation we understand not only the attainment of an approximate positioning of the OFDM symbols but also the detection of the OFDM signal itself. Such a distinction was not considered in the derivation of the ML timing estimators in §4.2. However, this distinction is of major importance when dealing with burst frame transmissions, where the receiver should reliably detect the presence of a frame. In the frame detection problem, the ramping-up characteristic of the AGC becomes the main disturbing issue.

The solution provided in [Mor99] for the frequency estimation looks very similar to the one in [Beek97]. Thus, if special symbols showing several periodicities are applied, they may be used not only for frequency but also for timing estimation. This idea was initially exploited in [Kel96] and [Schmidl97], for the case of having a symbol with two periods. As mentioned in §4.3.1, [Schmidl97] proposes the use of two symbols generated using PN sequences, whose elements are obtained from a QPSK constellation, but without including a cyclic prefix. In this context, the architecture shown in Figure 4.9 is proposed for the timing (θ) and the fine frequency (α) estimations. The preamble structure to be used is as in Figure 4.11, in which cyclic prefixes have been added to the preambles in Figure 4.8a. The corresponding results are shown in Figure 4.10 for the case of a noise-free OFDM system with N = 64, $N_G = 16$, $N_u = 52$, $\varepsilon = 1.4$ ($\alpha = -0.6$, $\beta = 2$). As it can be seen, the magnitude of the autocorrelation output J shows a very clear plateau of length N_G in the ideal case, which is derived from the cyclic extension of the first preamble symbol. Interestingly, the phase of J is also constant throughout the samples belonging to the plateau, since the only contribution to this phase is the one derived from the frequency offset. In Figure 4.11, where the timing correspondences are shown, it can be noticed that the very last sample of the plateau coincides with the beginning of the second preamble symbol, i.e. it provides the correct timing estimation θ. Hence, once the end of the plateau is detected, both time and frequency estimations may be acquired in one shot.

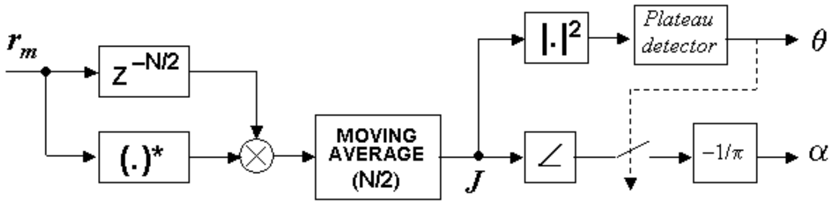


Figure 4.9. Architecture for joint time and fine frequency estimation based on a simplification of *van de Beek*'s solution [Beek97].

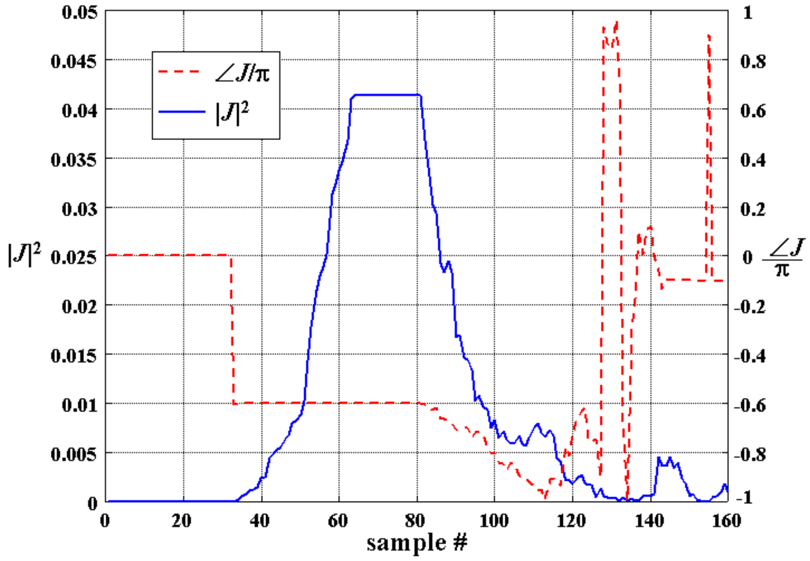


Figure 4.10. Magnitude and phase of J according to Figure 4.9.

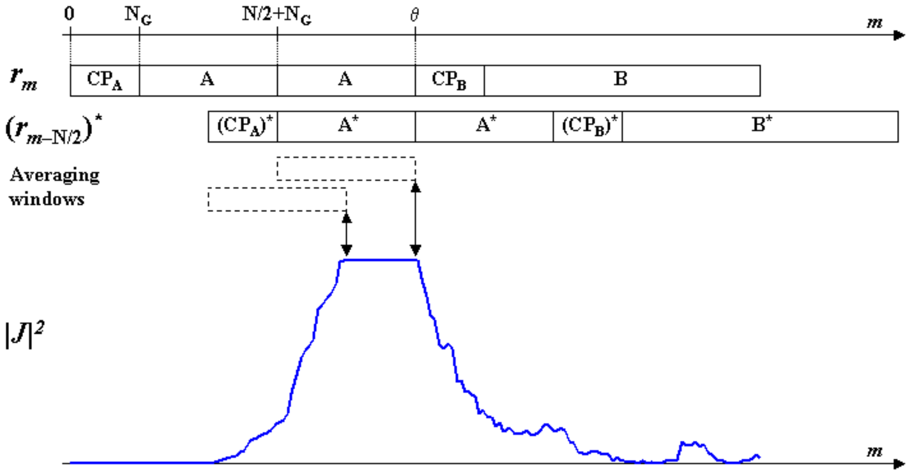


Figure 4.11. Time equivalences according to Figure 4.9.

Although simple, the scheme in Figure 4.9 was found to perform worse than the scheme presented in [Beek97] in a dispersive scenario, [Muel98]. In a multipath channel, the length of the plateau gets effectively reduced by N_m samples, N_m being the maximum dispersion in the channel, since the cyclic prefix is no longer an exact copy of the last portion of the symbol. Hence, the resulting timing acquisition is very unreliable and strongly depends on the channel. In §§ 4.3.3 and 4.3.4, methods to refine the timing estimation are going to be explored. Nevertheless, due to its simplicity and potential to increase the range of the frequency estimation, the architecture in Figure 4.9 is further investigated for implementing actual wireless OFDM systems, as it will be seen in Chapter 6, when dealing with the realization of a synchronizer for the IEEE 802.11a standard.

4.3.3 Refinement of the Timing Estimation

As mentioned in the previous section, the coarse timing estimation is not accurate enough to provide reliable symbol timing, but it just gives an indication of the existence of an incoming frame. If the remaining timing offset were not compensated, ISI and ICI may occur after FFT, thereby degrading the performance of the whole receiver severely. In order to refine the coarse timing the general solution consists in analyzing the information after FFT calculation, since timing errors will appear as a linear phase in addition to the ISI and ICI noise. A first approach in this direction was given in [Speth97], [Yang99a] and [Yang99b]. The main idea relies on the availability of pilot sub-channels inserted into the OFDM symbols, a method already suggested in [Class94] for the frequency estimation. In this case, however, no frequency offset should be present. Let's denote the original pilots inserted into the symbols as $C_{k,l}$ and the received pilots as $P_{k,l}$, with $k \in P$ being the set of P sub-carriers containing the pilots. In a multipath channel environment with time dispersion shorter than the cyclic prefix length, the channel is seen as a multiplicative complex coefficient in the frequency domain, i.e.

$$P_{k,l} = H_{k,l} \cdot C_{k,l} + V_{k,l} \, , \tag{4.45}$$

with $V_{k,l}$ being zero mean Gaussian noise.

The previous expression slightly varies when a timing offset θ is present. If the timing error is negative ($\theta<0$), (4.45) is further affected by a linear phase in the following form

$$P_{k,l} = C_{k,l} \cdot e^{j\frac{2\pi}{N}k\cdot\theta} \cdot H_{k,l} + V_{k,l} \, . \tag{4.46}$$

However, when the timing error is positive ($\theta>0$), (4.45) not only shows a linear phase, but ICI and ISI components appear due to the loss of orthogonality. In addition, the channel coefficient found on each sub-carrier is changed, i.e.

$$P_{k,l} = C_{k,l} \cdot e^{j\frac{2\pi}{N}k\cdot\theta} \cdot \left(H_{k,l} - H_{k,l}^{\theta}\right) + V_{k,l}^{\text{ISI+ICI}} + V_{k,l} \, . \tag{4.47}$$

A simple estimator for the channel coefficient may be given by

$$\hat{H}_{k,l} = \frac{P_{k,l}}{C_{k,l}} \tag{4.48}$$

for $k \in P$. One may estimate the channel impulse response (CIR) using the values obtained in (4.48) through an M-point IFFT operation (M < N). In this case, the timing offset will be visible as a shifting of the CIR after performing the IFFT. In order to easily recognize this shifting, the channel is supposed to have an exponentially decaying power delay profile or a strong line-of-sight component at zero delay. The procedure starts with the arrangement of the P available pilots into a vector $\hat{\mathbf{H}}$ of channel estimations following the *natural* order. By natural order it is meant the consideration of the positive and negative sub-carriers inside the OFDM symbol. If N sub-carriers are present, the ones corresponding to $k = 0,\ldots,(N/2)-1$ are defined as positive, whereas the ones corresponding to $k = N/2,\ldots,N-1$ are defined as negative, i.e. they are equivalent to $k = -N/2,\ldots,-1$. Thus, in the vector $\hat{\mathbf{H}}$ the pilots will be arranged starting from the most negative ones $(N/2 \leq k \leq N-1)$ and ending with the most positive ones $(0 \leq k < N/2)$.

The next step is a zero-padding operation on the vector $\hat{\mathbf{H}}$, therefore obtaining a new vector $\hat{\mathbf{H}}_\theta$ of dimensions M-by-1 as follows,

$$\hat{\mathbf{H}}_\theta = (\mathbf{W}_{1\mathrm{xM}})^{\mathrm{T}} \circ (\mathbf{0}_{1\mathrm{x}\frac{\mathrm{M-P}}{2}} \quad \hat{\mathbf{H}}_{1\mathrm{xP}} \quad \mathbf{0}_{1\mathrm{x}\frac{\mathrm{M-P}}{2}})^{\mathrm{T}}, \tag{4.49}$$

where **W** is a windowing sequence (Bartlett, Hamming,…) and the sign (○) states for an element-by-element vector multiplication. Now the IFFT of the vector in (4.49) is calculated in order to obtain the timing estimation. Depending on the expected resolution in the estimation, M is selected accordingly. This resolution is given by

$$\theta_{\mathrm{res}} = \frac{\mathrm{N}}{\Delta \cdot \mathrm{M}} \tag{4.50}$$

where Δ is the original pilot distance (in samples or sub-carriers).

A major drawback in this solution is that the pilots may be too much separated in frequency to be able to provide good channel estimations. As explained in [Troya01], the ideal value for the pilot separation is $\Delta = N/N_G$ in order to exploit the full cyclic prefix length. Any Δ bigger than this may produce channel estimations affected by aliasing, with undesirable effects on the timing estimation.

A second approach for the fine timing estimation deploys the solution proposed in [Schmidl97]. The scheme depicted in Figure 4.9 is able to obtain both, coarse timing and fine frequency offset, by means of the first preamble symbol only. In the same manner, the second preamble symbol may be used to obtain the fine timing estimation after being corrected for ε, as proposed in [Krstic03a]. The idea behind that is the fact that since the second preamble symbol is made up of a PN sequence with good correlation properties in the frequency domain, the same good properties should arise also in the time domain, since the IFFT is a linear operation. The results are shown in Figure 4.12, for the case N=64, N_u=52, N_G=16, SNR=10dB and a channel model A as given in Table D.1. The reference signal in the crosscorrelator is taken as the first N/2 samples of the second preamble symbol. As shown in Figure 4.12, a very clear peak is obtained at the output when the sequence stored in the crosscorrelator coincides with the reference.

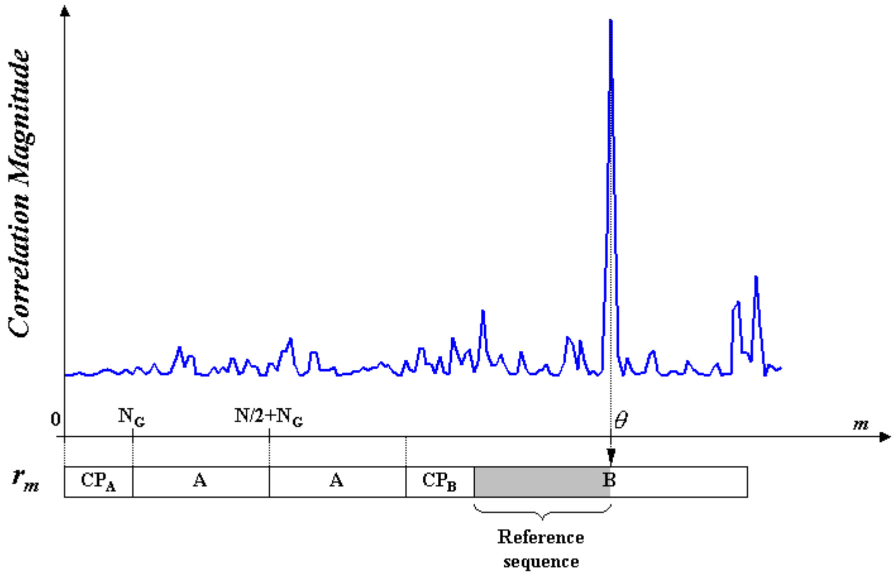


Figure 4.12. Results for the fine timing estimator based on the crosscorrelation with the second preamble symbol.

The previous method is not applicable when a big frequency offset is present, so it can only be applied once the second preamble symbol has been fully corrected for this offset. Still an uncertainty of $\pm T_S/2$ samples in the timing estimation will be present. This uncertainty will produce a residual linear phase on the successive symbols after FFT, which will be added to the linear phase derived from an impairment not considered until now in our synchronizer: the sampling clock frequency error. As analyzed in §3.4, the sampling clock mismatch is equivalent to a variable timing error. Therefore, after FFT calculation the sampling clock error appears as a linear phase whose slope changes on a symbol basis. This variable phase should be permanently monitored and tracked by some specific unit, as it will be shown in the next section.

4.3.4 Symbol Tracking

The classical ML solution investigated in §4.2.1 and shown in Figure 4.1 is able to track timing offsets through calculation of the error signal e_θ (the derivation is given in Appendix F). The authors in [Yang99b] propose an interesting method to generate such an error signal based on the pilot tones. For that, let's consider an approximation for the pilot at sub-channel *m* in symbol *l* to be as follows,

$$P_{m,l} \approx C_{m,l} \cdot H_{m,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot m\cdot\left(\mathrm{t}_\theta+\zeta\cdot\left(\mathrm{N_G}+l\cdot(\mathrm{N}+\mathrm{N_G})\right)\right)} + V^{\mathrm{I}}_{m,l} + V_{m,l}\,, \tag{4.51}$$

where t_θ represents the remaining timing offset and ζ is the sampling clock frequency offset expressed in ppm. The previous expression is the combination of the results obtained in (3.22) and (3.27), with the last two summands standing for the interference (ISI, ICI) and the AWGN noise components, respectively. It is further considered that the channel coefficients $H_{m,l}$ in (4.51) have been estimated by some means and their effect can be greatly compensated. If the channel does not change significantly from one symbol to the next one, the samples $H_{m,l}$ may be approximated by $\hat{H}_{m,l-1}$ as

$$\hat{H}_{m,l-1} \approx H_{m,l-1} \cdot e^{j\frac{2\pi}{\mathrm{N}}\cdot m\cdot\left(\hat{\mathrm{t}}_\theta+\zeta\cdot\left(\mathrm{N_G}+(l-1)\cdot(\mathrm{N}+\mathrm{N_G})\right)\right)} + V^{\mathrm{H}}_{m,l-1}\,, \tag{4.52}$$

where the second summand is a noise term which also includes the CTF estimation error.

In order to obtain a good discriminator for the timing offset, [Yang99b] suggests the use of two reference sequences, referred to as *early* and *late*, as follows

$$C_{m,l}^{early} \equiv C_{m,l} \cdot e^{j\frac{2\pi}{\mathrm{N}} \cdot m \cdot \delta}, \tag{4.53a}$$

$$C_{m,l}^{late} \equiv C_{m,l} \cdot e^{-j\frac{2\pi}{\mathrm{N}} \cdot m \cdot \delta}, \tag{4.53b}$$

with $0 < \delta < 1$ (samples).

The basic idea consists on comparing the pilots $P_{m,l}$ in (4.51) with these two sequences through a correlation, yielding

$$R_{early}(l) = \sum_{m \in P} \frac{P_{m,l}}{\hat{H}_{m,l-1}} \cdot (C_{m,l}^{early})^{*}, \tag{4.54a}$$

$$R_{late}(l) = \sum_{m \in P} \frac{P_{m,l}}{\hat{H}_{m,l-1}} \cdot (C_{m,l}^{late})^{*}. \tag{4.54b}$$

The actual value for δ depends on the maximum expected timing error inside the symbol. In our case, we may select δ to be 0.5 samples.

Now considering that the pilots are positioned at sub-carriers $m = p_0 + i \cdot \Delta$, with $0 \leq i \leq \mathrm{P}-1$ and Δ being the pilot distance, and inserting (4.51) and (4.52) into (4.54a/b), yields

$$R_{early}(l) \approx \sum_{i=0}^{\mathrm{P}-1} \left| C_{p_0+i\cdot\Delta,l} \right|^2 \cdot e^{j\frac{2\pi}{\mathrm{N}} \cdot (p_0+i\cdot\Delta) \cdot \left((\mathrm{t}_\theta - \hat{\mathrm{t}}_\theta) + \zeta \cdot (\mathrm{N}+\mathrm{N_G}) - \delta \right)} + V_{early}(l), \tag{4.55a}$$

$$R_{late}(l) \approx \sum_{i=0}^{\mathrm{P}-1} \left| C_{p_0+i\cdot\Delta,l} \right|^2 \cdot e^{j\frac{2\pi}{\mathrm{N}} \cdot (p_0+i\cdot\Delta) \cdot \left((\mathrm{t}_\theta - \hat{\mathrm{t}}_\theta) + \zeta \cdot (\mathrm{N}+\mathrm{N_G}) + \delta \right)} + V_{late}(l), \tag{4.55b}$$

where $V_{early}(l)$ and $V_{late}(l)$ are again noise components. The total timing error (in samples) is given by the expression

$$\Delta \mathrm{t} = (\mathrm{t}_\theta - \hat{\mathrm{t}}_\theta) + \zeta \cdot (\mathrm{N} + \mathrm{N_G}). \tag{4.56}$$

Therefore, the following error signal is generated using (4.55a) and (4.55b),

$$a(l, \Delta \mathrm{t}) = \left| R_{late}(l) \right|^2 - \left| R_{early}(l) \right|^2 = E\{a(l, \Delta \mathrm{t})\} + n_{a(l,\Delta \mathrm{t})}, \tag{4.57}$$

where $E\{.\}$ stands for the statistical expectation and $n_{a(l,\Delta \mathrm{t})}$ is a noise component. Hence, a timing discriminator can be defined as $S(\Delta \mathrm{t}, \delta) = E\{a(l, \Delta \mathrm{t})\}$. Note that $S(\Delta \mathrm{t}, \delta)$ is the result after filtering $a(l, \Delta \mathrm{t})$ by a low-pass filter. The expression for $S(\Delta \mathrm{t}, \delta)$ results as follows when $|C_{p0+i\cdot\Delta}|^2 = 1$ $(0 \leq i \leq \mathrm{P}-1)$,

$$\begin{aligned} S(\Delta \mathrm{t}, \delta) &= \left| \sum_{i=0}^{\mathrm{P}-1} e^{j\frac{2\pi}{\mathrm{N}} \cdot i \cdot \Delta \cdot (\Delta \mathrm{t} + \delta)} \right|^2 - \left| \sum_{i=0}^{\mathrm{P}-1} e^{j\frac{2\pi}{\mathrm{N}} \cdot i \cdot \Delta \cdot (\Delta \mathrm{t} - \delta)} \right|^2 \\ &= \left| \frac{\sin\left(\frac{\pi}{\mathrm{N}} \cdot \mathrm{P} \cdot \Delta \cdot (\Delta \mathrm{t} + \delta) \right)}{\sin\left(\frac{\pi}{\mathrm{N}} \cdot \Delta \cdot (\Delta \mathrm{t} + \delta) \right)} \right|^2 - \left| \frac{\sin\left(\frac{\pi}{\mathrm{N}} \cdot \mathrm{P} \cdot \Delta \cdot (\Delta \mathrm{t} - \delta) \right)}{\sin\left(\frac{\pi}{\mathrm{N}} \cdot \Delta \cdot (\Delta \mathrm{t} - \delta) \right)} \right|^2. \end{aligned} \tag{4.58}$$

The signal generated by the discriminator is used as the error signal in a Digital Phase Locked Loop (DPLL), as shown in Figure 4.13, which is very similar to the scheme provided in Figure 4.1. The normalized s-curve of the discriminator is depicted in Figure 4.14 for P = 4, $\Delta = 14$, $\delta = 0.5$. Unlike to the solution proposed in [Yang99b], where the timing correction was partly realized by controlling the sampling oscillator itself, the solution presented here is fully digital, i.e. the sampling is realized by a free running oscillator and any timing errors (including sampling clock errors) are corrected by digital means. The latter solution has several advantages: it allows to reduce the interaction between analog and digital circuitry; it allows to use cheaper analog components and its design is easier to handle as no joint analog and digital modeling technique has to be employed. Further details on the implementation of the Control and the Timing Error Compensation blocks in Figure 4.13 are provided in §7.3.4.

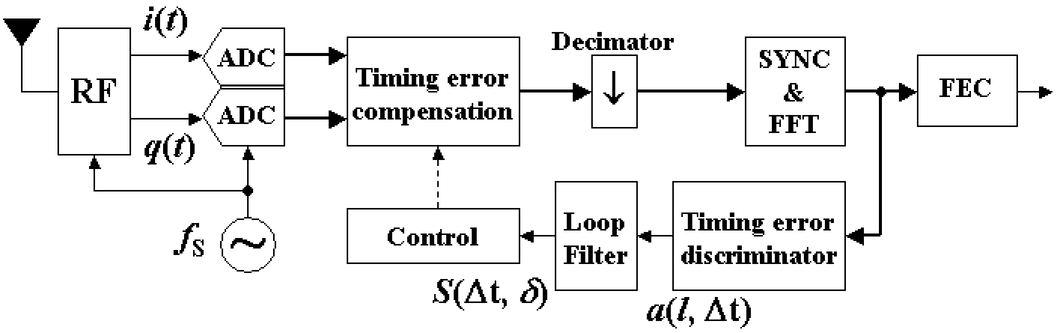


Figure 4.13. Simplified block diagram of a fully digital timing correction scheme.

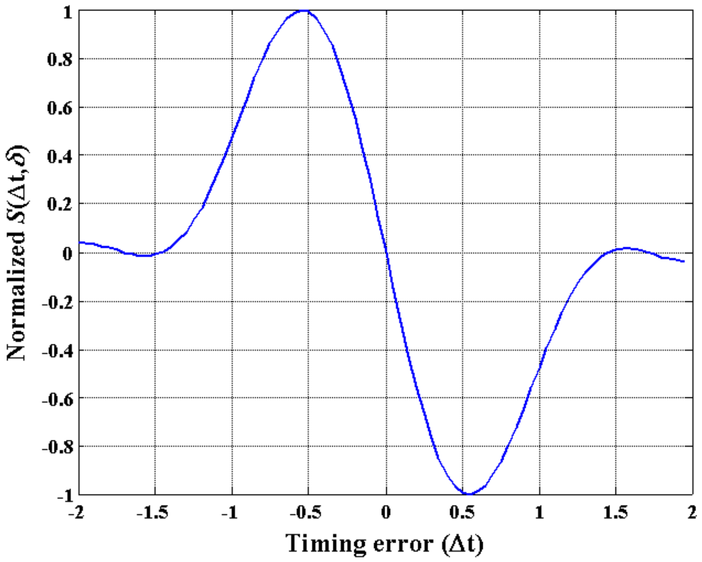


Figure 4.14. S-curve obtained from the pilot-based timing error discriminator as a function of the timing error (in samples) for P = 4, $\Delta = 14$, $\delta = 0.5$.

4.4 Summary

In the foregoing sections a thorough analysis of the most common acquisition strategies for normalized time (θ) and frequency (ε) offsets has been given, since these will be the parameters to be estimated by a synchronizer. In the case of the time acquisition, the natural range of estimation is $|\theta| < 0.5$ since discrete systems cannot go below this limit. As opposed to this, frequency estimators are supposed to provide an unlimited estimation range. Solutions based on both Non-Data Aided (NDA) as well as Data-Aided (DA) schemes have been investigated.

Firstly, the ML principle was applied in the determination of NDA synchronization schemes. The classical ML solution resulted in highly complex structures with a limited estimation range of the frequency offset ($|\varepsilon| < 0.5$). In addition, it was found that the timing estimator depends not only on the conforming pulse $g(t)$ but also on its derivative $\partial g(t)/\partial t$. Interestingly, the pulse $g'(t) = g(t)\cdot\partial g(t)/\partial t$ preserves the orthogonality, meaning that the resulting timing estimations are not disturbed by ICI.

A modified ML estimator exploiting the existence of the cyclic prefix in the OFDM symbols was further analyzed. The resulting synchronizer structure is significantly simpler compared to the classical ML, but the limitation in the frequency estimation range is still present.

Furthermore, the acquisition time for both time and frequency estimations is relatively large in the ML solutions, i.e. in the order of tens of symbols, restricting its application to systems with continuous transmission.

Increased estimation range with decreased acquisition time is only possible by providing special information within the OFDM symbols. Therefore, the so-called DA synchronizers are based on the utilization of specific preamble symbols. It has been seen that the frequency estimation range can be extended by generating a periodic pattern in the symbols. This property can also be used to acquire coarse timing estimation or frame detection. Nevertheless, in a multipath fading channel, the coarse timing estimation is not reliable enough.

Generally, preamble symbols are generated by PN sequences in the frequency domain. The good correlation properties are also maintained after conversion into the time domain. Hence, a crosscorrelator may be used to acquire fine timing estimation. Such a solution is only applicable if it is assured that no significant frequency offset is affecting the preamble symbol.

Finally, a tracking algorithm is required to permanently monitor the timing error since a sampling clock offset results in a variable timing. A simple solution based on the usage of pilot tones has been proposed. Nevertheless details on the realization of such a scheme have been deliberately omitted and will be given in Chapter 7, were a solution based on variable interpolation is provided.

The previous results regarding DA synchronization methods will be the basis for the development of an efficient synchronizer architecture for WLAN systems in Chapter 6.

Chapter 5

Channel Estimation in OFDM Systems

5.1 Introduction

In the foregoing chapter, the synchronization block was analyzed as one of the two main components of the *Inner Receiver*. The second block under consideration therein deals with the estimation and correction of the *filtering* affecting the OFDM signal. This filtering is mainly due to the multipath transmission channel found in wireless communications, but the several filters located in the transceiver hardware play an important role as well. As a result, the OFDM symbols are extended in time by an amount equal to the summation of the impulse response lengths of all the filters involved in the transmission and reception chain. Such an extension provokes the leakage of a symbol into the successive one, resulting in Inter-Symbol Interference. As analyzed in Appendix C, one interesting feature of the OFDM signals is their capacity to overcome ISI when cyclically extending each symbol by a number of samples N_G (or T_G seconds). This has two main advantages: on one hand, the possible leakage from the previous symbol is fully absorbed as long as it is shorter than the cyclic extension. On the other side, the examination of the OFDM symbols in the frequency domain (after DFT) arises to be much more convenient, since now the filtering appears inside the OFDM symbols as multiplicative complex factors affecting each of the sub-carriers. In view of this fact, the *equalization*, i.e. channel correction, becomes much easier since it can be realized by means of a complex division in the frequency domain. This is of big help if compared to the traditional single-carrier transmission schemes, where the equalization is realized using complicated transversal filter structures (see [Pro95], Chapter 10).

The CTF inside the OFDM symbols is estimated following three possible strategies. The first approach found in the literature is based on the existence of pilot sub-carriers (or pilot tones) and was firstly proposed by Cimini in [Cim85]. The main idea lies on the fact that both, transmitter and receiver, know the information carried onto the pilot tones. Thus, the multiplicative channel factors corresponding to the pilots can be easily extracted, and the channel coefficients belonging to the data sub-carriers may then be obtained by interpolation of the pilot tones. Consequently, the Sampling Theorem has to be considered carefully when deciding the position of the pilots inside the OFDM symbols.

In those OFDM systems employing a small number of sub-carriers, the use of pilots happens to be a big burden since the spectral efficiency of the system is drastically reduced. In this case, the CTF is estimated based on a *decision-directed* procedure in which *virtual pilots* are obtained from the demodulated/decoded information.

A last strategy found in the literature deals with the so-called blind and semi-blind estimation methods, which are in general less used since they require a much higher computational complexity. These methods are not going to be considered in this Dissertation.

5.2 Pilot-assisted Channel Estimators

5.2.1 Conditions on the Pilot Grid

In §2.2.2, four statistical parameters describing a multipath fading channel were presented. On one side, the *maximum excess delay*, T_m, corresponds to the maximum dispersion found in the channel, which at the same time provides an approximation of the channel bandwidth. The *channel coherence bandwidth* B_c is typically given as $B_c \approx 1/T_m$. If two sinusoids with frequency separation greater than B_c are transmitted through the channel, this will affect them in a substantially different way, i.e. the attenuation coefficients affecting each of the sinusoids will be very low correlated with each other.

In an OFDM system, each sub-carrier should “percept” the channel as being frequency non-selective, i.e. the CTF should exhibit a nearly constant behavior inside the sub-carrier bandwidth. This is achieved by allowing the sub-carrier spacing Δf to be much smaller than B_c. In addition, as pilot tones are used to interpolate the CTF, the channel coefficients corresponding to consecutive pilots should be enough correlated. In order to find a suitable value for the pilot separation, the Sampling Theorem is taken as the reference. Considering that the pilot separation is given as $N_F \cdot \Delta f$, the Sampling Theorem states that

$$N_F \cdot \Delta f \leq \frac{B_c}{2} . \tag{5.1}$$

On the other hand, from the Doppler power spectrum of the channel a *maximum Doppler spread* B_d is derived. This parameter is directly related to the relative speed existing between the transmitter and the receiver and is given in (2.5). The inverse of B_d is called the *coherence time* T_d, and represents the mean time during which the channel does not change significantly. Let's consider that two pilot tones are placed exactly at the same frequency position, but separated by N_L OFDM symbols (see Figure 5.1). These two pilots may be used to estimate the CTF at the same frequency position of the symbols falling in between if their corresponding channel coefficients were enough correlated. Remembering that the symbol timing is given as $T+T_G$, the Sampling Theorem is applied in order to decide the maximum separation between pilots in the time direction, i.e. $N_L \cdot (T+T_G)$, yielding

$$N_L \cdot (T + T_G) \leq \frac{T_d}{2} . \tag{5.2}$$

Aiming to decrease the complexity of the interpolators, the authors in [Hoe97] recommend to oversample the CTF in both time and frequency directions by a factor of 2, resulting in the following expressions

$$\mathrm{N_F} \cdot \Delta f \cdot \mathrm{T_m} \approx \frac{1}{4}, \tag{5.3a}$$

$$\mathrm{N_L} \cdot (\mathrm{T} + \mathrm{T_G}) \cdot \mathrm{B_d} \approx \frac{1}{4}. \tag{5.3b}$$

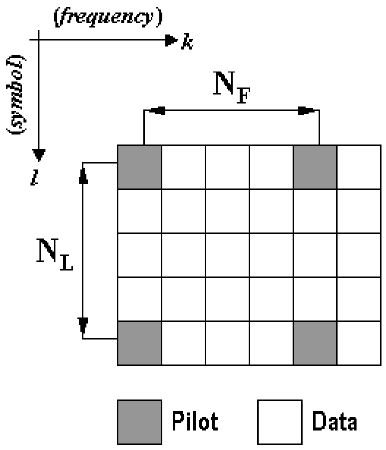


Figure 5.1. Definition of pilot spacing in the time and frequency directions.

There is still another restriction on the pilot spacing in the frequency direction $\mathrm{N_F}$. Provided that a certain cyclic prefix of length $\mathrm{N_G}$ is considered in the OFDM symbols, the maximum excess delay allowed for the channel, $\mathrm{N_m}$ (in samples, considering a sampling time T/N), should be smaller than $\mathrm{N_G}$. Expression (3.9) was relating the CIR with the CTF. Let's consider for the time being that the different channel delays τ_i are multiples of the sampling interval T/N and let's express (3.9) using the following matrix notation

$$\mathbf{H}_l = \mathbf{F}^{\mathrm{H}} \cdot \begin{pmatrix} \mathbf{h}_l \\ 0 \end{pmatrix}. \tag{5.4}$$

The vectors $\mathbf{h}_l$ and $\mathbf{H}_l$ contain the samples of the CIR and the CTF for symbol l, respectively, and have dimension N-by-1. The matrix $\mathbf{F}^{\mathrm{H}}$ defines the DFT and has elements

$$W_{k,n}^{*} = (\sqrt{\mathrm{N}})^{-1} \cdot e^{-j\frac{2\pi}{\mathrm{N}}kn}, \tag{5.5}$$

with $0 \le k, n \le \mathrm{N}-1$.

The case being treated here considers the knowledge of P equidistant elements (pilot tones) in matrix $\mathbf{H}_l$, with a distance between the pilots as in (5.3a). If the pilots are placed at positions $k_0, k_1, \ldots, k_{\mathrm{P}-1}$, (5.4) results as follows

$$\begin{pmatrix} H_{k_0} \\ H_{k_1} \\ \vdots \\ H_{k_{\mathrm{P}-1}} \end{pmatrix} = \underbrace{\frac{1}{\sqrt{\mathrm{N}}} \begin{pmatrix} 1 & e^{-j\frac{2\pi}{\mathrm{N}} \cdot k_0 \cdot 1} & \cdots & e^{-j\frac{2\pi}{\mathrm{N}} \cdot k_0 \cdot (\mathrm{N_G}-1)} \\ 1 & e^{-j\frac{2\pi}{\mathrm{N}} \cdot k_1 \cdot 1} & \cdots & e^{-j\frac{2\pi}{\mathrm{N}} \cdot k_1 \cdot (\mathrm{N_G}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j\frac{2\pi}{\mathrm{N}} \cdot k_{\mathrm{P}-1} \cdot 1} & \cdots & e^{-j\frac{2\pi}{\mathrm{N}} \cdot k_{\mathrm{P}-1} \cdot (\mathrm{N_G}-1)} \end{pmatrix}}_{\mathbf{Q}^{\mathrm{H}}} \cdot \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{\mathrm{N_G}-1} \end{pmatrix}, \tag{5.6}$$

which represents a system of P equations with N_G unknown variables. Thus, the condition $P = N_G$ should be satisfied in order to obtain a unique solution for the vector **h**. In this case, the matrix **Q** is a Vandermonde matrix with the following general form

$$\mathbf{Q} = \mathrm{Q}(x_0,...,x_{K-1}) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{K-1} & x_1^{K-1} & \cdots & x_{K-1}^{K-1} \end{pmatrix}. \tag{5.7}$$

According to [Gol96], the matrix **Q** is non-singular as long as the x_k are distinct and therefore it is invertible.

Note that from what it has been stated previously, the optimal number of pilot tones in the frequency domain has been derived as well as the maximum spacing between those pilots. In [Negi98] the authors demonstrate that in an AWGN scenario, the Mean Square Error (MSE) of any interpolator is minimized if all the pilots inside the OFDM symbol are equidistant. Nevertheless, the exact position of the pilots can be freely chosen. In a multi-path scenario, some sub-carriers may be strongly attenuated by the channel. If any of these strongly disrupted sub-carriers corresponds to a pilot tone the interpolation error will increase dramatically. In order to overcome this problem a rhombus type grid is typically used, as shown in Figure 5.2.

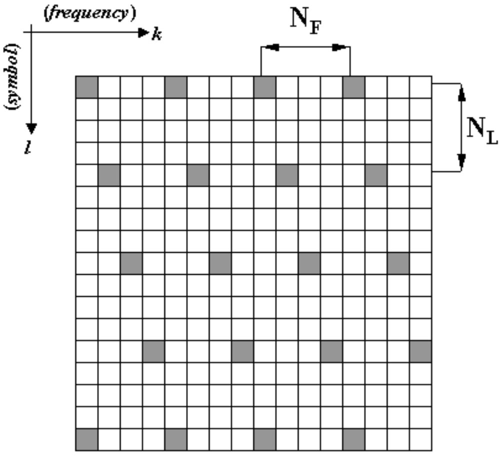


Figure 5.2. Pilot grid with rhombus type distribution. This arrangement diminishes degradation of channel estimation when one pilot tone is severely attenuated by the channel.

5.2.2 Common Interpolation Methods

Considering that the pilot spacing inside the OFDM symbols is as in (5.3a/b), several interpolation strategies can be followed in order to obtain the channel coefficient in the data sub-carriers. Strictly speaking, the interpolation problem shown in Figure 5.1 is two-dimensional (2D). As a first approach, one may try to interpolate the values in the data sub-carriers applying 2D *sinc* functions. As shown in [Pro96], *sinc* functions are the ideal functions to be used for interpolation. In a first step, the Least Square (LS) estimations of the channel coefficients at each pilot tone should be obtained by doing

$$\hat{H}_{k',l'} = \frac{P_{k',l'}}{C_{k',l'}} = H_{k',l'} + \frac{V_{k',l'}}{C_{k',l'}} \tag{5.8}$$

with $(k', l') \in \mathcal{P}$ being the set of sub-carriers k' and symbols l' containing the pilots, $C_{k',l'}$ being the expected value for the particular pilot on symbol l' and sub-carrier k', $P_{k',l'}$ the actual received value for that pilot and $V_{k',l'}$ the samples of a zero-mean Gaussian noise process. In a second step, a 2D signal should be generated placing the LS channel estimates at their correct positions and setting all the other values to zero. This signal is therefore filtered through an interpolation filter with a general impulse response given by

$$H_{\mathrm{INT}}(t_0, t_1) = \frac{\sin\left(\pi \cdot (t_0^2 + t_1^2)\right)}{\pi \cdot (t_0^2 + t_1^2)}. \tag{5.9}$$

The vector $(t_0\ t_1)$ is rotated according to the selected sampling grid in order to provide the desired vector (k,l) as follows

$$\begin{pmatrix} k \\ l \end{pmatrix} = \mathbf{G} \cdot \begin{pmatrix} t_0 \\ t_1 \end{pmatrix}. \tag{5.10}$$

The matrix **G** is called the *sampling matrix* [Vaid93]. In case of using a rectangular grid, the transformation in (5.10) results

$$\begin{pmatrix} k \\ l \end{pmatrix} = \begin{pmatrix} \mathrm{N_F} & 0 \\ 0 & \mathrm{N_L} \end{pmatrix} \cdot \begin{pmatrix} t_0 \\ t_1 \end{pmatrix} \tag{5.11}$$

whereas, a grid as the one shown in Figure 5.2 would result in the following transformation

$$\begin{pmatrix} k \\ l \end{pmatrix} = \begin{pmatrix} \mathrm{N_F} & \Delta \\ 0 & \mathrm{N_L} \end{pmatrix} \cdot \begin{pmatrix} t_0 \\ t_1 \end{pmatrix}, \tag{5.12}$$

with Δ=1 in that particular example. Substituting t_0 and t_1 by k and l in (5.9) results in the following interpolation filter for the rectangular grid (see Figure 5.3a),

$$H_{\mathrm{INT}}(k,l) = \frac{\sin\left(\pi \cdot \left(\left(\frac{k}{\mathrm{N_F}}\right)^2 + \left(\frac{l}{\mathrm{N_L}}\right)^2\right)\right)}{\pi \cdot \left(\left(\frac{k}{\mathrm{N_F}}\right)^2 + \left(\frac{l}{\mathrm{N_L}}\right)^2\right)} \tag{5.13}$$

whereas, for the rhombus type grid the filter has the following impulse response (see Figure 5.3b)

$$H_{\mathrm{INT}}(k,l) = \frac{\sin\left(\pi \cdot \left(\left(\frac{k}{\mathrm{N_F}}\right)^2 + \left(1 + \frac{\Delta^2}{\mathrm{N_F^2}}\right) \cdot \left(\frac{l}{\mathrm{N_L}}\right)^2 - \frac{2\Delta}{\mathrm{N_L N_F^2}} k \cdot l\right)\right)}{\pi \cdot \left(\left(\frac{k}{\mathrm{N_F}}\right)^2 + \left(1 + \frac{\Delta^2}{\mathrm{N_F^2}}\right) \cdot \left(\frac{l}{\mathrm{N_L}}\right)^2 - \frac{2\Delta}{\mathrm{N_L N_F^2}} k \cdot l\right)}. \tag{5.14}$$

An important parameter to be also considered is the so-called *sampling density* ρ_S, which is defined in [Vaid93] as

$$\rho_\mathrm{S} = \frac{1}{\left|\det(\mathbf{G})\right|} \tag{5.15}$$

Both grids considered in expressions (5.11) and (5.12) result in the same sampling density $\rho_\mathrm{S} = (\mathrm{N_F} \cdot \mathrm{N_L})^{-1}$, meaning that the rhombus type grid still complies with the Sampling Theorem.

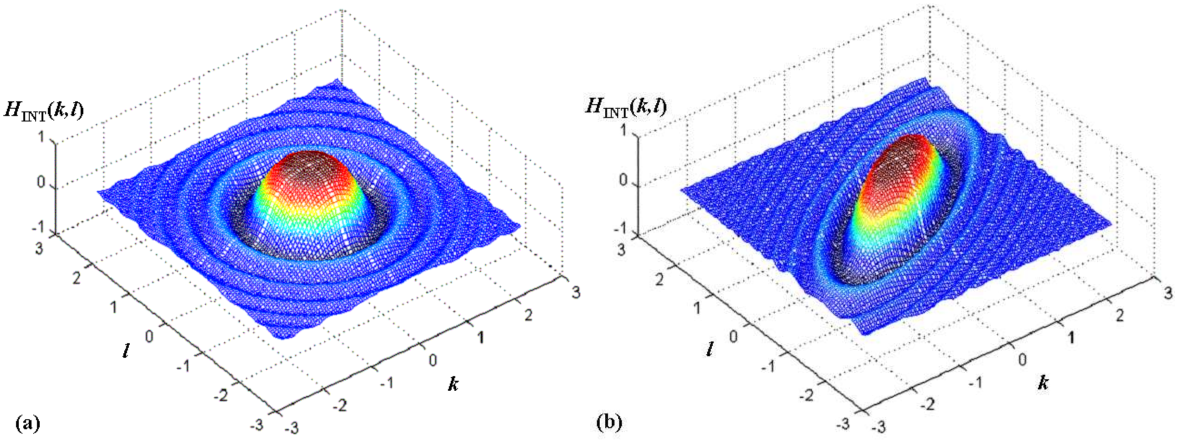


Figure 5.3. Two-dimensional *sinc* functions with: a) $N_F = 1$, $N_L = 1$, $\Delta=0$; b) $N_F = 1$, $N_L = 1$, $\Delta=1$.

One major drawback of the ideal interpolation is that the estimation of the channel coefficient at position (k,l) depends on the LS estimates obtained from a huge number of pilots, i.e. the cardinality of the set $\mathcal{P}$, $\|\mathcal{P}\|$, comprises all the transmitted pilots in the whole OFDM frame. In view of a practical implementation, each tuple (k,l) is assigned a limited number of pilots from a sub-set $\mathcal{T}_{k,l}$ of $\mathcal{P}$ ($\mathcal{T}_{k,l} \subset \mathcal{P}$) following, when possible, a symmetric pattern. A direct consequence of this is the degradation of the mean squared error (MSE) of the channel estimations. Moreover, as it can be seen from Figure 5.2, those sub-carriers placed at the band edges cannot be assigned a symmetric interpolation pilot pattern, leading to a further degradation of the MSE. Further simplifications may be achieved if other interpolation functions than *sinc* functions are used. As an example, the authors in [Moon00] make a comparison of linear, Gaussian and cubic spline interpolation filters for the special case of the DAB standard. However, the interpolation filters seen so far are not optimal in the sense that they are not designed to minimize the MSE of the channel estimations. Since the LS estimates obtained from expression (5.8) are affected by the noise, a more general definition of the interpolation problem is needed, as it will be seen in the following subsection.

5.2.3 Minimum Mean-Squared Error (MMSE) Interpolation Methods

In order to find the interpolator minimizing the MSE, the following general expression defining the interpolation operation needs to be considered,

$$\tilde{H}_{k,l} = \sum_{(k',l')\in\mathcal{T}_{k,l}} w_{k,l,k',l'} \cdot \hat{H}_{k',l'}\,, \tag{5.16}$$

where the $\hat{H}_{k',l'}$ are the LS estimates obtained from the pilots as in (5.8), $w_{k,l,k',l'}$ are the weighting factors associated to the tuple (k,l) from pilot (k',l'), and $\tilde{H}_{k,l}$ is the final channel estimation for the tuple (k,l). Note that expression (5.16) defines a two-dimensional filter. Our purpose is the minimization of the mean squared value of the estimation error given by

$$e_{k,l} = H_{k,l} - \tilde{H}_{k,l}\,, \tag{5.17}$$

with the mean squared error defined as

$$J_{k,l} = E\{|e_{k,l}|^2\}\,. \tag{5.18}$$

The optimal filter in the sense of minimizing $J_{k,l}$ with the MMSE criterion is the 2-D Wiener filter [Hay02]. The filter coefficients of the 2-D Wiener filter are obtained by applying the orthogonality principle in linear mean squared estimation [Hay02]. This results in the following expression [Kai98, pp. 89-90]

$$E\{e_{k,l} \cdot \hat{H}^{*}_{k'',l''}\} = 0, \quad \forall(k'',l'') \in \mathcal{T}_{k,l} . \tag{5.19}$$

The orthogonality principle states that the mean squared error $J_{k,l}$ is minimum if the filter coefficients $w_{k,l,k',l'}$ $\forall(k',l') \in \mathcal{T}_{k,l}$, are selected such that the error $e_{k,l}$ is orthogonal to all LS estimates $\hat{H}_{k'',l''}$ $\forall(k'',l'') \in \mathcal{T}_{k,l}$. The orthogonality principle leads to the Wiener-Hopf equation [Kai98] given by

$$E\{H_{k,l}\hat{H}^{*}_{k'',l''}\} = \sum_{(k',l')\in\mathcal{T}_{k,l}} w_{k,l,k',l'} \cdot E\{\hat{H}_{k',l'}\hat{H}^{*}_{k'',l''}\}, \quad \forall(k'',l'') \in \mathcal{T}_{k,l} , \tag{5.20}$$

which is obtained by substituting (5.16) and (5.17) into (5.19).

The left side term in (5.20) represents a crosscorrelation function,

$$\theta_{k-k'',l-l''} = E\{H_{k,l}\hat{H}^{*}_{k'',l''}\} . \tag{5.21}$$

Now, considering (5.8) and by assuming that $V_{k'',l''}$ has zero mean and is statistically independent of the pilot tones $C_{k'',l''}$, the crosscorrelation function in (5.21) is equal to the discrete time-frequency correlation function

$$\theta_{k-k'',l-l''} = E\{H_{k,l}H^{*}_{k'',l''}\} . \tag{5.22}$$

The time-frequency correlation function corresponds to the Fourier transform of the power delay profile of the channel, i.e. $\mathcal{F}_{\tau}\{\phi_{h}(\Delta t;\tau)\}$ -see (2.4)-.

The right side term in (5.20) includes an autocorrelation function given by

$$\phi_{k'-k'',l'-l''} = E\{\hat{H}_{k',l'}\hat{H}^{*}_{k'',l''}\} , \tag{5.23}$$

which can be further written in the following form considering (5.8),

$$\phi_{k'-k'',l'-l''} = \theta_{k'-k'',l'-l''} + \frac{1}{\gamma_{C}}\delta_{k'-k'',l'-l''} , \tag{5.24}$$

where γ_C states for the SNR and $\delta_{k'-k'',l'-l''}$ represents the Kronecker delta function. However, in the previous expression it is further assumed that all the transmitted symbols have the same average power, i.e. the pilot tones are not boosted.

Inserting (5.21) and (5.23) into (5.20), yields in vector notation

$$\boldsymbol{\theta}^{\mathrm{T}}_{k,l} = \boldsymbol{\Phi} \cdot \mathbf{w}^{\mathrm{T}}_{k,l} \tag{5.25}$$

The matrix $\boldsymbol{\Phi}$ is the $\mathrm{N_{tap}}$-by-$\mathrm{N_{tap}}$ pilot autocorrelation matrix and $\boldsymbol{\theta}_{k,l}$ is the crosscorrelation vector of length $\mathrm{N_{tap}}$, where $\mathrm{N_{tap}}$ equals the cardinality of the sub-set $\mathcal{T}_{k,l}$, i.e. $\mathrm{N_{tap}} \doteq \|\mathcal{T}_{k,l}\|$. The vector $\mathbf{w}_{k,l}$ of length $\mathrm{N_{tap}}$ contains the weighting factors $w_{k,l,k',l'}$ required to obtain the estimate $\tilde{H}_{k,l}$. Hence, the filter coefficients of the optimum 2-D Wiener filter are

$$\mathbf{w}^{\mathrm{T}}_{k,l} = \boldsymbol{\Phi}^{-1} \cdot \boldsymbol{\theta}^{\mathrm{T}}_{k,l} , \tag{5.26}$$

when assuming that the auto- and crosscorrelation functions are perfectly known and $\mathrm{N_{tap}} = \|\mathcal{P}\|$, i.e. all the available pilot tones inside the frame are used to estimate each single channel coefficient corresponding to the data tones.

The first reference mentioning the use of 2-D Wiener filters in OFDM systems was given by P. Hoeher in [Hoe91] and later on in [Hoe97]. In the mobile radio channel, it can be assumed that the power delay profile, $\phi_h(\Delta t;\ \tau)$, and the Doppler power spectrum, $S_H(\lambda;\Delta f\,)$, are statistically independent [Hoe92] –the relationship between these two functions is given in Figure 2.4 by the function $\phi_H(\Delta t;\ \Delta f)$–. This fact enables us to rewrite expression (5.22) considering that $\phi_H(\Delta t;\ \Delta f) = \phi_H(\Delta t)\cdot\phi_H(\Delta f)$ as follows,

$$\theta_{k-k'',l-l''} = E\{H_{k,l}H^*_{k'',l''}\} = E\{H_k H_l H^*_{k''} H^*_{l''}\} = \theta_{\mathrm{K}}(k-k'')\theta_{\mathrm{L}}^{\mathrm{T}}(l-l'') \tag{5.27}$$

with

$$\theta_{\mathrm{K}}(k-k'') = \phi_{\mathrm{H}}(\Delta f)\Big|_{\Delta f=(k-k'')\frac{1}{\mathrm{T}}} = \int_{-\infty}^{\infty}\phi_{\mathrm{h}}(\tau)\cdot e^{-j2\pi\tau\Delta f}\,d\tau\Bigg|_{\Delta f=(k-k'')\frac{1}{\mathrm{T}}} \tag{5.28.a}$$

$$\theta_{\mathrm{L}}(l-l'') = \phi_{\mathrm{H}}(\Delta t)\Big|_{\Delta t=(l-l'')\cdot(\mathrm{T}+\mathrm{T_G})} = \int_{-\infty}^{\infty}\mathrm{S_H}(\lambda)\cdot e^{j2\pi\lambda\Delta t}\,d\lambda\Bigg|_{\Delta t=(l-l'')\cdot(\mathrm{T}+\mathrm{T_G})} \tag{5.28.b}$$

The final values for the vectors $\theta_K(k–k'')$ and $\theta_L(l–l'')$ will depend on the considered channel model. The power delay profile is generally taken as a normalized exponentially decaying function as shown in Figure 2.2a. This normalization is valid in the interval $0 \le \tau \le T_G$, yielding the following expression for $\phi_h(\tau)$,

$$\phi_{\mathrm{h}}(\tau) = \frac{1}{\tau_{\mathrm{rms}}\left(1-e^{\frac{-\mathrm{T_G}}{\tau_{\mathrm{rms}}}}\right)}e^{\frac{-\tau}{\tau_{\mathrm{rms}}}}, \quad \tau \ge 0 \tag{5.29}$$

where the parameter τ_{rms} is selected according to the allowed maximum excess delay $T_m = T_G$, i.e. the cyclic prefix length. Since the function in (5.29) has semi-infinite support, an appropriate way to determine τ_{rms} may be as follows,

$$e^{\frac{-\mathrm{T_G}}{\tau_{\mathrm{rms}}}} = \delta \quad \Rightarrow \quad \tau_{\mathrm{rms}} = \frac{-\mathrm{T_G}}{\ln(\delta)}, \tag{5.30}$$

with δ being an arbitrary small positive value, $\delta << 1$.

Now inserting (5.29) into (5.28a) and integrating from 0 to T_G, results in the following expression for the frequency correlation function

$$\theta_{\mathrm{K}}(k-k'') = \frac{\left(1-e^{-\mathrm{T_G}\left(\frac{1}{\tau_{\mathrm{rms}}}+j2\pi\cdot((k-k'')/\mathrm{T})\right)}\right)}{\left(1-e^{\frac{-\mathrm{T_G}}{\tau_{\mathrm{rms}}}}\right)\left(1+j2\pi\cdot(k-k'')\cdot(\tau_{\mathrm{rms}}/\mathrm{T})\right)}. \tag{5.31}$$

In order to determine the time correlation function in (5.28b), the expression given in (2.6) for the Doppler power spectrum is rewritten here in a normalized form, i.e.

$$\mathrm{S_H}(\lambda) = \begin{cases} \dfrac{1}{\pi \cdot \mathrm{B_d}} \left(1 - \left(\dfrac{\lambda}{\mathrm{B_d}} \right)^2 \right)^{\frac{-1}{2}}; & -\mathrm{B_d} \le \lambda \le \mathrm{B_d} \\ 0; \quad \text{otherwise} \end{cases} \tag{5.32}$$

By considering the definition of the zeroth order Bessel function of the first kind [Bron74] as

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} e^{j \cdot x \cdot \cos\alpha} d\alpha \,, \tag{5.33}$$

the integral in (5.28b) can be solved by inserting (5.32) and making the change of variable $\lambda/\mathrm{B_d} = \cos(\alpha)$, yielding

$$\theta_{\mathrm{L}}(l - l'') = J_0(2\pi \cdot \mathrm{B_d} \cdot (l - l'') \cdot (\mathrm{T} + \mathrm{T_G})) \,. \tag{5.34}$$

The previous results are valid for any kind of grid, either rectangular or rhombus-type.

5.2.4 Separable Interpolation Filters

The two-dimensional filtering treated in the foregoing sections becomes computationally intractable as the number of pilot tones considered in the analysis increases. Interestingly, [Hoe91] points out the possibility of decoupling the 2-D Wiener filter into two cascaded 1-D filters. In this case, both filters can be designed for their worst-case points of operation, i.e. allowed maximum excess delay, maximum Doppler frequency and worst SNR, which leads to fixed (pre-computable) filter coefficients. The principle is shown in Figure 5.4 for the case of a rectangular grid. Firstly, a filter is applied in the frequency direction to all those OFDM symbols containing pilot tones. The estimated channel coefficients for the data bearing tones (represented as ⊠ in Figure 5.4) are then considered as *pseudo-pilots* in the time direction filter.

It can be readily noticed that all the pilots employed in the frequency direction filter have $(l'–l'') = 0$. If $\mathrm{K_F}$ taps are taken in this filter, the pilot autocorrelation matrix $\mathbf{\Phi_F}$ will result as follows

$$\mathbf{\Phi}_{\mathrm{F}} = \begin{pmatrix} \theta_{\mathrm{K}}(0) + \dfrac{1}{\gamma_{\mathrm{C}}} & \theta_{\mathrm{K}}(-\mathrm{N_F}) & \cdots & \theta_{\mathrm{K}}(-\mathrm{N_F} \cdot (\mathrm{K_F} - 1)) \\ \theta_{\mathrm{K}}(\mathrm{N_F}) & \theta_{\mathrm{K}}(0) + \dfrac{1}{\gamma_{\mathrm{C}}} & \cdots & \theta_{\mathrm{K}}(-\mathrm{N_F} \cdot (\mathrm{K_F} - 2)) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{\mathrm{K}}(\mathrm{N_F} \cdot (\mathrm{K_F} - 1)) & \theta_{\mathrm{K}}(\mathrm{N_F} \cdot (\mathrm{K_F} - 2)) & \cdots & \theta_{\mathrm{K}}(0) + \dfrac{1}{\gamma_{\mathrm{C}}} \end{pmatrix} . \tag{5.35}$$

Similarly, all pilots employed in the time direction filter have $(k'–k'') = 0$. Considering $\mathrm{K_L}$ taps, the pilot autocorrelation matrix $\mathbf{\Phi_L}$ yields

$$\mathbf{\Phi}_{\mathrm{L}} = \begin{pmatrix} \theta_{\mathrm{L}}(0) + \dfrac{1}{\gamma_{\mathrm{C}}} & \theta_{\mathrm{L}}(-\mathrm{N_L}) & \cdots & \theta_{\mathrm{L}}(-\mathrm{N_L} \cdot (\mathrm{K_L} - 1)) \\ \theta_{\mathrm{L}}(\mathrm{N_L}) & \theta_{\mathrm{L}}(0) + \dfrac{1}{\gamma_{\mathrm{C}}} & \cdots & \theta_{\mathrm{L}}(-\mathrm{N_L} \cdot (\mathrm{K_L} - 2)) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{\mathrm{L}}(\mathrm{N_L} \cdot (\mathrm{K_L} - 1)) & \theta_{\mathrm{L}}(\mathrm{N_L} \cdot (\mathrm{K_L} - 2)) & \cdots & \theta_{\mathrm{L}}(0) + \dfrac{1}{\gamma_{\mathrm{C}}} \end{pmatrix} . \tag{5.36}$$

The final values of the coefficients in the frequency and time directions, $w_F(k–k')$ and $w_L(l–l')$, depend on the selected groups of pilots.

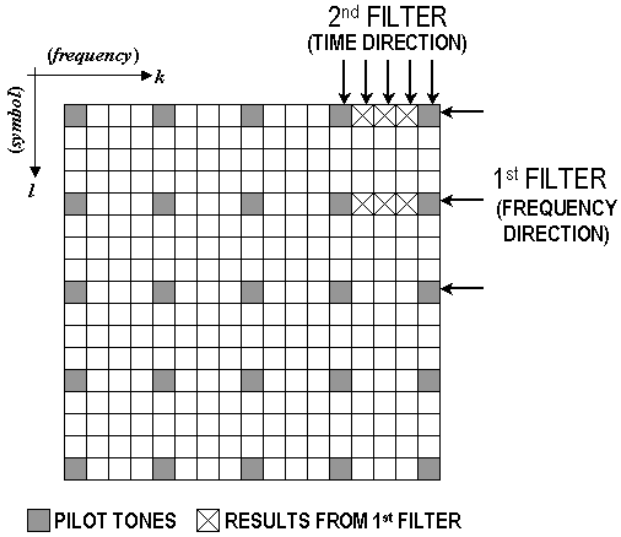


Figure 5.4. Principle of 2x1-D FIR filtering.

For the case of the sinc filters, the separation into 2x1-D filters is straightforward. The coefficients in the frequency direction are decided upon the following expression

$$w_{k-k',0} = \frac{\sin\left(\pi \cdot \frac{(k-k')}{\mathrm{N_F}}\right)}{\pi \cdot \frac{(k-k')}{\mathrm{N_F}}}, \tag{5.37}$$

where k is the position of the sub-carrier and k' the position of the pilot under consideration. Equivalently, the time direction filter shows the following coefficients for the sub-carriers on symbol l, when the pilots (or *pseudo-pilots*) on the symbols l' are considered

$$w_{0,l-l'} = \frac{\sin\left(\pi \cdot \frac{(l-l')}{\mathrm{N_L}}\right)}{\pi \cdot \frac{(l-l')}{\mathrm{N_L}}}. \tag{5.38}$$

The results in (5.26), (5.37) and (5.38) will be used in §5.2.6 to simulate the MSE of the 2D and 2x1D Wiener filters.

5.2.5 The Low-Rank Approach

In §5.2.3 the Minimum Mean Squared Error (MMSE) interpolator for the CTF has been derived for the particular case in which pilot tones are available. Theoretically, this solution should provide us with the least noisy channel estimations among all the possible estimators for a given number of pilot tones. Nevertheless, this is only true if a big amount of pilots are available. As it will be seen in §5.2.6, when comparing the different channel interpolators, the *edge effect* rather than the noise will be the limiting factor in the Wiener solution. Thus, some alternative method for noise reduction needs to be devised.

Considering the channel estimation under another perspective, one may recall the expression given in (5.4) relating CTF and CIR. This expression is rewritten here for the noisy scenario,

$$\mathbf{H}_l = \mathbf{F}^{\mathrm{H}} \cdot \left[\begin{pmatrix} \mathbf{h}_l \\ \mathbf{0} \end{pmatrix} + \mathbf{v}_l \right] = \mathbf{H}_l^{ideal} + \mathbf{V}_l \,, \tag{5.39}$$

where $\mathbf{F}^{\mathrm{H}}$ represents the matrix defining the N-point DFT, $\mathbf{h}_l$ is the vector containing the N_G samples of the CIR, the vector $\mathbf{0}$ is the $(N-N_G)$-by-1 all-zero vector and $\mathbf{v}_l$ contains the N samples of a zero-mean white Gaussian noise process with power σ_v^2. The transformed vector $\mathbf{V}_l$ is also zero-mean Gaussian with the same power σ_v^2 since it is the result of a linear unitary transformation of the vector $\mathbf{v}_l$.

However, only a limited number of equidistant samples $P < N$ is known from the vector $\mathbf{H}_l$ in (5.39), i.e. the pilot tones, which are obtained according to expression (5.8). The procedure starts with the arrangement of the P available pilots into a new vector $\hat{\mathbf{H}}_l$ of channel estimations in which the data tones are forced to 0 whereas, the pilot tones are substituted by their corresponding noisy LS estimations. The N-point vector with the estimation of the CIR, $\hat{\mathbf{h}}_l$, can be obtained from the vector $\hat{\mathbf{H}}_l$ as follows,

$$\hat{\mathbf{h}}_l = \mathbf{F} \cdot \hat{\mathbf{H}}_l \,. \tag{5.40}$$

Due to the particular structure of the vector $\hat{\mathbf{H}}_l$, the estimated vector $\hat{\mathbf{h}}_l$ will show a periodic structure with period P samples. Since the vector $\hat{\mathbf{H}}_l$ has P out of N non-zero equidistant samples, the noise content in $\hat{\mathbf{h}}_l$ is also reduced by a factor N/P. Therefore, the smaller P is, the bigger is the noise reduction, though P is lower bounded by the expected maximum excess delay for the channel.

For the final channel estimation only the first N_G samples of $\hat{\mathbf{h}}_l$ are required, while the rest of samples is forced to be 0, yielding

$$\tilde{\mathbf{H}}_l = \mathbf{F}^{\mathrm{H}} \cdot \mathbf{W} \cdot \mathbf{F} \cdot \hat{\mathbf{H}}_l \,, \tag{5.41}$$

where $\mathbf{W}$ represents an N-by-N square matrix with the form

$$\mathbf{W} = \begin{pmatrix} \mathbf{I}_{N_G \times N_G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{5.42}$$

The procedure described above is shown schematically in Figure 5.5 and is referred to as *low-rank* channel estimation. The concept behind the low rank channel estimators was firstly exposed in [Beek95a], and further refined in [Edf96] and [Edf98]. In [Edf96], the authors suggest to apply a number of weighting factors to the estimation $\hat{\mathbf{h}}_l$ in (5.40) prior to DFT calculation, modifying the matrix $\mathbf{W}$ as follows

$$\mathbf{W} = \begin{pmatrix} \mathbf{D}_{N_G \times N_G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{5.43}$$

where the N_G-by-N_G matrix $\mathbf{D}$ is purely diagonal. The elements of the main diagonal δ_i, $i = 0, \ldots, N_G-1$, are derived from the knowledge of the SNR (γ_C) and the channel PDP sampled at intervals $i \cdot (T/N)$, yielding

$$\delta_i = \frac{\phi_h(i)}{\phi_h(i) + \dfrac{1}{\gamma_C}} \,. \tag{5.44}$$

Note that when a high SNR scenario is considered, the solutions for $\mathbf{W}$ in (5.42) and (5.43) converge, i.e. no weighting is necessary.

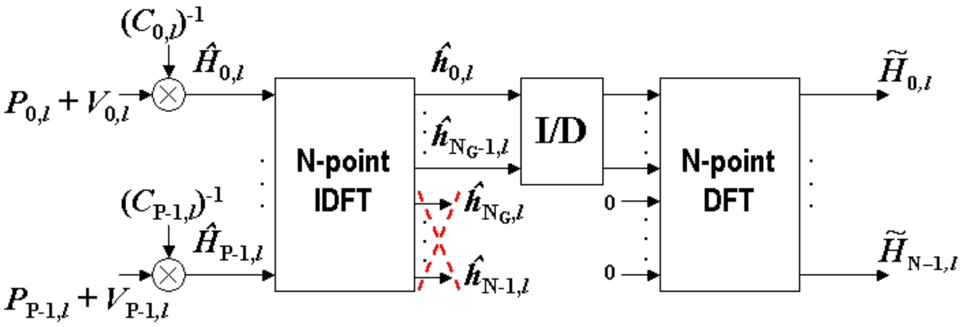


Figure 5.5. Channel Transfer Function estimation through a low-rank approximation.

Although the concept of low-rank estimation has been specified in the frequency direction, it may also be applied in the time direction [Sandell96]. In this case a sufficient number of pilots taken at the same frequency position from different symbols has to be collected. The minimum amount of pilots necessary depends on the expected time variation rate of the channel.

As a closing remark, it is worthy to note that the low-rank method can only be applied as long as no linear phase error appears in the frequency domain (see method exposed in §4.3.3 for time error estimation). It is well known that a linear phase in the frequency domain is equivalent to a circular shift in the time domain sequence. Hence, the obtained CIR in Figure 5.5 may be significantly shifted and the zero insertion may finally result in the elimination of important paths.

5.2.6 Mean-Squared Error Performance of Interpolators

In order to evaluate the performance of the previously described interpolation methods, an OFDM system with 64 sub-carriers was taken into consideration. The main parameters are given in Table 5.1. The system makes use of 54 out of 64 sub-carriers. One of every sixth symbols contains 10 pilots placed at positions #-27, #-21, #-15, #-9, #-3, #3, #9, #15, #21, #27. In the simulations, frames containing 37 symbols where generated. With this pilot configuration and frame length the overall edge effect is reduced since: a) the pilot bearing symbols contain a pilot tone at each edge of the band, and b) both the first and last symbols in the frame are pilot bearing symbols.

The channel model A as given in Table D.1 was considered in the simulations. The minimum tap delay defined for this channel is 10 ns, which is 5 times shorter than the sample spacing defined for the OFDM signal. Hence, the following modifications were made:

1. The OFDM symbols were interpolated by a factor of 8 using zero padding and a 512-point IFFT. This reduced the sample spacing to be 6.25 ns
2. The delays defined for each tap in the channel model were approximated to the nearest multiple of 6.25 ns.

In Figures 5.6/5.7 the results obtained from the Wiener interpolators are shown for the 2-D and 2x1-D cases, respectively. Perfect knowledge of the SNR and the channel statistics is assumed. Surprisingly, these results reveal that the edge effect plays a key role when such a small number of sub-carriers is used.

Furthermore, the MSE obtained for high SNR values is worse than that for low SNR. This is because the channel noise is able to randomize the edge effects.

Table 5.1. Parameters of the OFDM system used in the comparison of the different interpolation methods.

RF bandwidth	20 MHz
Carrier frequency	5.6 GHz
Number of OFDM symbols in the frame	37
Number of sub-carriers (N)	64
OFDM symbol duration (T)	3.2 μs
Cyclic prefix duration (T_G)	400 ns
Total symbol duration ($T+T_G$)	3.6 μs
Sub-carrier spacing (Δf)	312.5 kHz
Total number of used sub-carriers per symbol (N_u)	54
Pilot symbol distance in frequency (N_F)	6
Pilot symbol distance in time (N_L)	6
Expected maximum excess delay in the channel (T_m)	400 ns
Expected maximum mobile speed	3 m/s
Trials per MSE calculation	40

Figures 5.8/5.9 show the MSE obtained for the interpolators based on the sinc function. In Figure 5.8 it can be seen that an increasing number of filter taps in the 2-D solution is counterproductive, since the edge effects become much more visible. Significant improvement is achieved in this case by the separation of the 2-D filter into two cascaded 1-D filters (Figure 5.9). With the parameters selected for this particular example, there seems to be an optimum value for the number of taps on each filter for both low SNR and high SNR conditions. In the former, only the inclusion of two taps on each filter is recommended whereas for the later, three taps are suggested.

Finally, the results obtained for the low-rank approach are shown in Figure 5.10. For the case in which weighting was applied, a perfect knowledge of the channel statistics as well as the SNR was considered. Since one of every sixth symbol contains pilots, the low-rank method is applied to the pilot bearing symbols and the resulting CTF is kept constant for the next five symbols. It can be seen from Figure 5.10 that the inclusion of the weighting parameters does not improve significantly the performance of the channel estimator. Furthermore, the results are very similar to the ones shown in Figure 5.9 for the 2x3 taps case. This is not surprising since these two strategies are similar, i.e. the whole transformation of the signal into the time domain and then again into the frequency domain with some samples forced to zero (low-rank approach), is equivalent to a convolution with a sinc function in the frequency domain.

Nevertheless, the use of pilots has to be seen as a burden to the system since the spectral efficiency is reduced. Therefore, alternative solutions avoiding the use of any pilot tones should be considered. A possible scheme is now analyzed in §5.3.

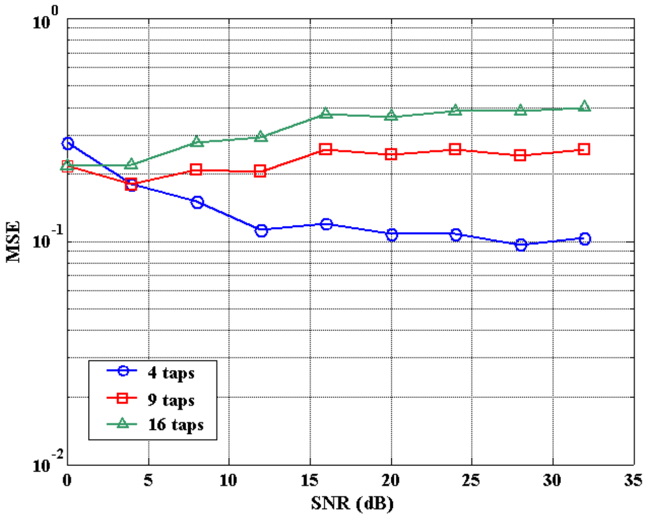


Figure 5.6. Average mean squared error (relative to the channel power) for the 2-D Wiener filter.

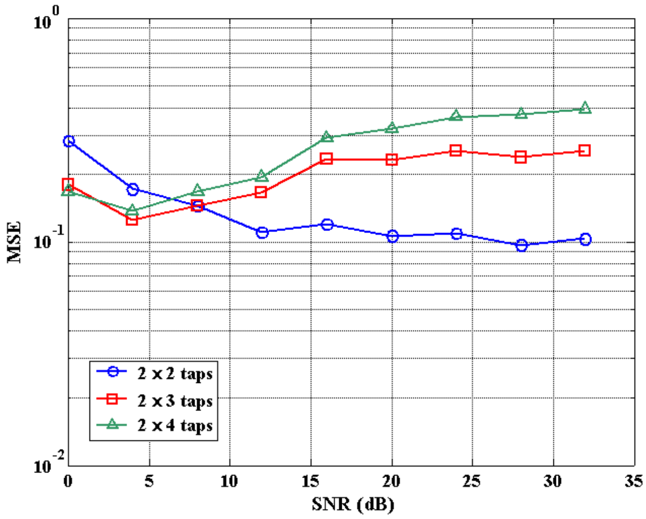


Figure 5.7. Average mean squared error (relative to the channel power) for the 2x1-D Wiener filters.

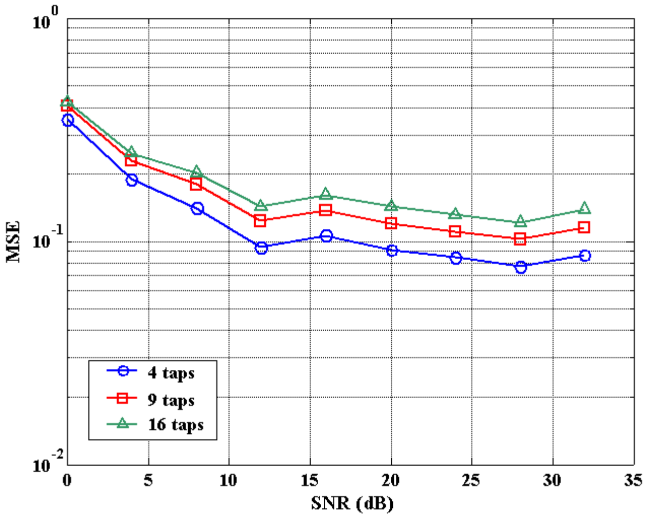


Figure 5.8. Average mean squared error (relative to the channel power) for the 2-D Sinc filter.

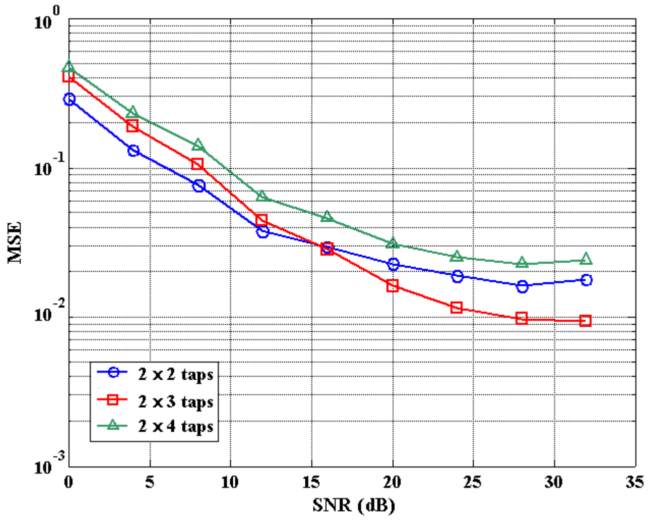


Figure 5.9. Average mean squared error (relative to the channel power) for the 2x1-D Sinc filters.

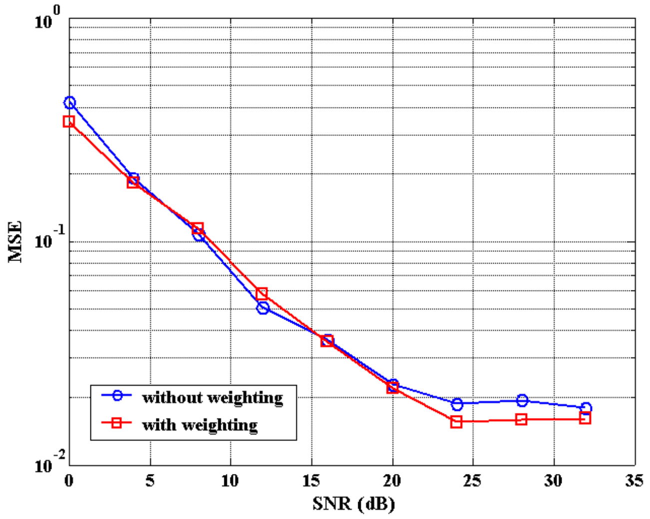


Figure 5.10. Average mean squared error (relative to the channel power) for the 1-D Low-rank approach.

5.3 Decision-Directed Channel Estimators

5.3.1 Principles of Operation

In those OFDM systems employing a small number of sub-carriers, the use of pilots happens to be a big burden since the spectral efficiency of the system is drastically reduced. In this case, the CTF is estimated based on a *decision-directed* procedure in which *virtual pilots* are obtained from the demodulated/decoded information. Hence, no pilot tones need to be inserted into the OFDM symbols. Mignone and Morello initially proposed the Coded Decision-Directed Demodulation (CD3) scheme depicted in Figure 5.11 [Mig96]. The processing is done in the following steps:

1. The whole procedure starts with the inclusion of a reference symbol, which is transmitted first in the OFDM frame. The content of this symbol is generally a BPSK modulated PN sequence selected in such a way that the PAPR is minimized. Since this sequence is known at the receiver side, a reference estimation for the CTF can be easily obtained by applying a sign correction.

The corrected sequence is filtered to reduce its noise content and stored as the correct channel estimation.

2. The stored reference is applied to the next incoming symbol (first data symbol) in the block denoted as *Equalizer* in Figure 5.11 in order to correct for the channel attenuation on the different sub-carriers. The obtained sequence $\tilde{X}_{k,l}$ is then demodulated, deinterleaved and decoded by means of a soft Viterbi algorithm. At the same time the uncorrected symbol $Y_{k,l}$ is stored in the *Symbol Buffer* block (Figure 5.11).
3. This is the key step of the whole procedure. The output bits from the Viterbi algorithm, $b_{k,l}$, are fed back into the *Channel Estimator* block. This block applies an encoding, interleaving and modulation to the bit sequence $b_{k,l}$, thus obtaining an estimation of the data $\hat{X}_{k,l}$. In the absence of any noise, the sequences $\tilde{X}_{k,l}$ and $\hat{X}_{k,l}$ should be exactly the same. At this point, the sequence $\hat{X}_{k,l}$ is considered to be the correct transmitted sequence and treated as a set of *pseudo-pilots* in order to obtain a new estimation for the CTF.
4. The new CTF, $\hat{H}_{k,l}$, is attained as the division of the sequence $Y_{k,l}$ by $\hat{X}_{k,l}$ (Least Squares estimation). The estimation noise present in $\hat{H}_{k,l}$ is reduced by the frequency and time direction filters, which will be explained in more detail in §§5.3.2 and 5.3.3, respectively.
5. The final CTF achieved after filtering, $\tilde{H}_{k,l}$, is taken as the CTF to be used for correction of the next data symbol. At this point, the procedure is repeated from step 2 onwards until the end of the frame is reached.

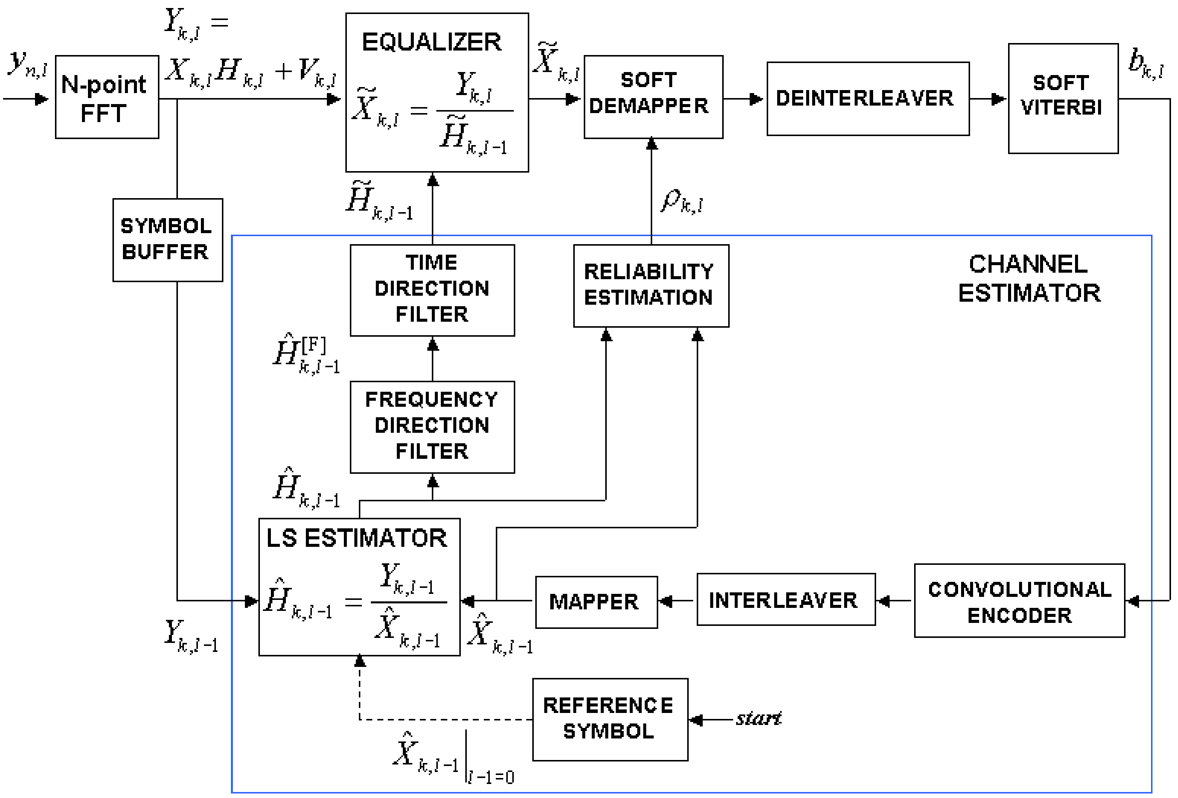


Figure 5.11. Block diagram of the CD3 scheme according to Mignone & Morello [Mig96].

Since the solution shown in Figure 5.11 is a feedback solution, it is highly sensitive to the noise induced through the feedback loop. Hence, the CD3 scheme exploits the noise reduction capabilities of the Forward Error Correction (FEC) in order to increase its performance. Nevertheless, this noise reduction is not enough to guarantee a reliable performance and additional methods should be considered. Thus, both frequency and time direction filters are proposed in the next subsections in order to reduce the variance of the channel estimations.

5.3.2 Frequency Direction Filter

The frequency direction filter is based on the solution proposed in §5.2.5 for the low-rank approach. However, in the present context (5.41) has to be modified considering that now N_u out of N sub-carriers, i.e. the data-carrying tones, are used as *pseudo-pilots*. The difference $N_v = N–N_u$ corresponds to the guard band and the DC component. Our objective is to determine a close form for the matrix $\mathbf{\Theta_N} = \mathbf{F^H} \cdot \mathbf{W} \cdot \mathbf{F}$, with $\mathbf{W}$ given as in (5.42), considering the particular distribution of the data-carrying tones. A simple method to obtain the *noise reduction matrix* $\mathbf{\Theta_N}$ was initially addressed in [Schmidt01]. Generally, the OFDM symbol obtained after DFT calculation in Figure 5.11 is organized as shown in Table 5.2, were N_u is considered to be a multiple of 2.

The channel estimations $\hat{H}_{k,l}$ in Figure 5.11 may be arranged in a vector form as $\hat{\mathbf{H}}_l^-$ for the $N_u/2$ samples corresponding to the negative sub-carriers, $\hat{\mathbf{H}}_l^+$ for the $N_u/2$ samples corresponding to the positive ones and $\hat{\mathbf{H}}_l^{\varnothing}$ for the remaining $N–N_u$ non-used samples, yielding

$$\hat{\mathbf{H}}_l = \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \\ \hat{\mathbf{H}}_l^{\varnothing} \end{pmatrix}. \tag{5.45}$$

At the same time, the vector representing the CIR, $\hat{\mathbf{h}}_l$, can be decomposed into two components, $\hat{\mathbf{h}}_{1,l}$ and $\hat{\mathbf{h}}_{0,l}$, corresponding to the first N_G non-zero samples and the last $N–N_G$ zero (or noisy) samples, respectively, i.e.

$$\hat{\mathbf{h}}_l = \begin{pmatrix} \hat{\mathbf{h}}_{1,l} \\ \hat{\mathbf{h}}_{0,l} \end{pmatrix}. \tag{5.46}$$

Expressions (5.45) and (5.46) are related by the following transformation,

$$\begin{pmatrix} \hat{\mathbf{h}}_{1,l} \\ \hat{\mathbf{h}}_{0,l} \end{pmatrix} = \begin{pmatrix} \mathbf{F_{11}} & \mathbf{F_{12}} \\ \mathbf{F_{21}} & \mathbf{F_{22}} \end{pmatrix} \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \\ \hat{\mathbf{H}}_l^{\varnothing} \end{pmatrix}, \tag{5.47}$$

where $\mathbf{F_{11}}$ is a N_G-by-N_u matrix, $\mathbf{F_{12}}$ is a N_G-by-$(N–N_u)$ matrix, $\mathbf{F_{21}}$ is a $(N–N_G)$-by-N_u matrix and $\mathbf{F_{22}}$ is a $(N–N_G)$-by-$(N–N_u)$ matrix. The elements of these matrices are all of the form

$$W_{n,k} = (\sqrt{N})^{-1} \cdot e^{j\frac{2\pi}{N} \cdot n \cdot k}, \tag{5.48}$$

where *n* stands for the row value and *k* for the column value. Hence, the following values for *n* and *k* must be considered in the generation of the matrices $\mathbf{F_{11}}$, $\mathbf{F_{12}}$, $\mathbf{F_{21}}$ and $\mathbf{F_{22}}$ (values for *k* are derived straightforward from Table 5.2):

- Matrix $\mathbf{F}_{11} \Rightarrow n \in [0, N_G-1]$ (corresponding to rows 1 to N_G); $k \in [N-(N_u/2), N-1] \cup [1, N_u/2]$ (corresponding to columns 1 to N_u).
- Matrix $\mathbf{F}_{12} \Rightarrow n \in [0, N_G-1]$ (corresponding to rows 1 to N_G); $k \in [0] \cup [(N_u/2)+1, N-(N_u/2)-1]$ (corresponding to columns 1 to $N-N_u$).
- Matrix $\mathbf{F}_{21} \Rightarrow n \in [N_G, N-1]$ (corresponding to rows 1 to $N-N_G$); $k \in [N-(N_u/2), N-1] \cup [1, N_u/2]$ (corresponding to columns 1 to N_u).
- Matrix $\mathbf{F}_{22} \Rightarrow n \in [N_G, N-1]$ (corresponding to rows 1 to $N-N_G$); $k \in [0] \cup [(N_u/2)+1, N-(N_u/2)-1]$ (corresponding to columns 1 to $N-N_u$).

Table 5.2. Correspondences between DFT sample out and sub-channel number.

DFT sample out	0	1	2	3	...	$\frac{N_u}{2}$	$\frac{N_u}{2}+1$	...	$N-\frac{N_u}{2}-1$	$N-\frac{N_u}{2}$	...	N-2	N-1
Sub-channel #	DC	+1	+2	+3	...	$+\frac{N_u}{2}$	Ø	...	Ø	$-\frac{N_u}{2}$	...	-2	-1

From (5.47) one may write $\hat{\mathbf{h}}_{0,l}$ as

$$\hat{\mathbf{h}}_{0,l} = \mathbf{F}_{21} \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \end{pmatrix} + \mathbf{F}_{22} \cdot \hat{\mathbf{H}}_l^{\varnothing}\,. \tag{5.49}$$

But our design constraint states that $\hat{\mathbf{h}}_{0,l} \doteq \mathbf{0}$, yielding

$$\mathbf{F}_{22} \cdot \hat{\mathbf{H}}_l^{\varnothing} = -\mathbf{F}_{21} \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \end{pmatrix}. \tag{5.50}$$

Now it is possible to isolate $\hat{\mathbf{H}}_l^{\varnothing}$ in the previous expression. For that, the authors in [Schmidt01] define a modified pseudo-inverse matrix of $\mathbf{F}_{22}$, i.e. $\mathbf{F}_{22}^{+}$, as follows,

$$\mathbf{F}_{22}^{+} = (\mathbf{F}_{22}^{\mathrm{H}} \cdot \mathbf{F}_{22} + \gamma^2)^{-1} \cdot \mathbf{F}_{22}^{\mathrm{H}}, \tag{5.51}$$

where γ is a dummy parameter, $0 < \gamma << (1/N)$, used to prevent possible numerical instability in the matrix inversion. By this, $\hat{\mathbf{H}}_l^{\varnothing}$ in (5.50) results as follows,

$$\hat{\mathbf{H}}_l^{\varnothing} = -\mathbf{F}_{22}^{+} \cdot \mathbf{F}_{21} \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \end{pmatrix}, \tag{5.52}$$

which is an estimation of the values of the CTF at those sub-carriers where no information is available, based on the knowledge of the CTF at the sub-carriers where information is transmitted.
The previous result can be applied in (5.47) to obtain an expression for $\hat{\mathbf{h}}_{1,l}$, i.e.

$$\hat{\mathbf{h}}_{1,l} = \mathbf{F}_{11} \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \end{pmatrix} + \mathbf{F}_{12} \cdot \hat{\mathbf{H}}_l^{\varnothing} = (\mathbf{F}_{11} - \mathbf{F}_{12} \cdot \mathbf{F}_{22}^{+} \cdot \mathbf{F}_{21}) \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^- \\ \hat{\mathbf{H}}_l^+ \end{pmatrix}. \tag{5.53}$$

Once the key matrices have been identified, now the new vector $\hat{\mathbf{H}}_l^{[F]}$ resulting from the low-rank method has to be derived considering the matrix **W** given in (5.42) -no weighting version-. The new vector can be written as follows considering the inverse transformation of expression (5.47),

$$\hat{\mathbf{H}}_l^{[\mathrm{F}]} = \begin{pmatrix} (\hat{\mathbf{H}}_l^{[\mathrm{F}]})^{-} \\ (\hat{\mathbf{H}}_l^{[\mathrm{F}]})^{+} \\ (\hat{\mathbf{H}}_l^{[\mathrm{F}]})^{\varnothing} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_{11}^{\mathrm{H}} & \mathbf{F}_{21}^{\mathrm{H}} \\ \mathbf{F}_{12}^{\mathrm{H}} & \mathbf{F}_{22}^{\mathrm{H}} \end{pmatrix} \cdot \begin{pmatrix} \hat{\mathbf{h}}_{1,l} \\ \mathbf{0} \end{pmatrix} \tag{5.54}$$

and finally, introducing (5.53) into (5.54), yields

$$\begin{pmatrix} (\hat{\mathbf{H}}_l^{[\mathrm{F}]})^{-} \\ (\hat{\mathbf{H}}_l^{[\mathrm{F}]})^{+} \end{pmatrix} = \mathbf{F}_{11}^{\mathrm{H}} \cdot \hat{\mathbf{h}}_{1,l} = \mathbf{F}_{11}^{\mathrm{H}} \cdot (\mathbf{F_{11}} - \mathbf{F_{12}} \cdot \mathbf{F_{22}^{+}} \cdot \mathbf{F_{21}}) \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^{-} \\ \hat{\mathbf{H}}_l^{+} \end{pmatrix} = \mathbf{\Theta_N} \cdot \begin{pmatrix} \hat{\mathbf{H}}_l^{-} \\ \hat{\mathbf{H}}_l^{+} \end{pmatrix}, \tag{5.55}$$

where the Noise Reduction Matrix (NRM) $\mathbf{\Theta_N}$ can be pre-computed offline once the parameters N, $\mathrm{N_u}$ and $\mathrm{N_G}$ have been defined. The SNR gain (in dB) achieved by the frequency direction filter is given by

$$\upsilon_{\mathrm{dB}} = 10 \cdot \log_{10}\left(\frac{\mathrm{N}^2}{\mathrm{N_G} \cdot \mathrm{N_u}} \right), \tag{5.56}$$

considering that no error occurs in the estimation of the transmitted data, i.e. $\hat{X}_{k,l} = X_{k,l}$, and that the data have normalized power and are statistically independent of the noise. As an example, in the 802.11a standard the parameters in (5.56) are as follows: N = 64, $\mathrm{N_G} = 16$, $\mathrm{N_u} = 52$. This results in a SNR improvement of $\upsilon_{\mathrm{dB}} \approx 7$ dB due to the noise reduction matrix.

5.3.3 Time Direction Filter

The low-rank method used for the frequency domain filter may also be applied to reduce the noise in the time direction. For that, a number of symbols (not necessarily N) have to be collected and the method applied $\mathrm{N_u}$ times, once for each sub-carrier position containing information. Although being the optimal solution, due to its computational complexity, this method only makes sense if the channel has a big variation rate (high mobile speed scenario). A simpler approach was firstly suggested in [Negi98] for the case of slow varying fading channels. Under this scenario, the variation of the CTF in the time direction can be modeled as a first order Markov process as given in [Hay02], yielding

$$\mathbf{H}_{l+1} = a \cdot \mathbf{H}_l + \boldsymbol{\omega}_l, \tag{5.57}$$

where the vector $\boldsymbol{\omega}_l$ contains the samples of a noise process and $a < 1$ ($a \approx 1$) is a fixed parameter. Considering the model used until now for the fading channel (see §2.2), each sample in the vector $\boldsymbol{\omega}_l$ corresponds to a zero-mean normalized complex Gaussian stationary process. Furthermore, the samples in $\boldsymbol{\omega}_l$ are uncorrelated with each other and with the measured noise $\mathbf{V}_l$. The autocorrelation matrix of the vector $\boldsymbol{\omega}_l$ is given as $\mathbf{R}_{\boldsymbol{\omega}} = E\{\boldsymbol{\omega}_l \cdot \boldsymbol{\omega}_l^{\mathrm{H}}\} = \sigma_\omega^2 \cdot \mathbf{I}$.

The channel estimation vector obtained from the frequency direction filter, $\hat{\mathbf{H}}_l^{[\mathrm{F}]}$, undergoes a first-order low-pass IIR filter given by

$$\tilde{\mathbf{H}}_l = \kappa \cdot \tilde{\mathbf{H}}_{l-1} + (1-\kappa) \cdot \hat{\mathbf{H}}_l^{[\mathrm{F}]}, \tag{5.58}$$

where κ represents a *forgetting factor*. In order to find the optimum value for κ, the tap error vector is defined as

$$\mathbf{e}_l = \tilde{\mathbf{H}}_l - \mathbf{H}_l, \tag{5.59}$$

with its autocorrelation matrix given as $\mathbf{R_e}(l) = E\{\mathbf{e}_l \cdot \mathbf{e}_l^{\mathrm{H}}\}$.

Expanding expression (5.59), yields

$$\begin{aligned}\mathbf{e}_l &= \kappa\cdot\tilde{\mathbf{H}}_{l-1}+(1-\kappa)\cdot\hat{\mathbf{H}}_l^{[\mathrm{F}]}-\mathbf{H}_l\\ &=\kappa\cdot\tilde{\mathbf{H}}_{l-1}+(1-\kappa)\cdot(\mathbf{H}_l+(\sqrt{\upsilon})^{-1}\cdot\mathbf{V}_l)-\mathbf{H}_l\\ &=\kappa\cdot\tilde{\mathbf{H}}_{l-1}-\kappa\cdot(a\cdot\mathbf{H}_{l-1}+\boldsymbol{\omega}_{l-1})+(1-\kappa)\cdot(\sqrt{\upsilon})^{-1}\cdot\mathbf{V}_l\\ &=\kappa\cdot\tilde{\mathbf{H}}_{l-1}-\kappa\cdot\mathbf{H}_{l-1}+\kappa\cdot\mathbf{H}_{l-1}-\kappa\cdot a\cdot\mathbf{H}_{l-1}-\kappa\cdot\boldsymbol{\omega}_{l-1}+(1-\kappa)\cdot(\sqrt{\upsilon})^{-1}\cdot\mathbf{V}_l\\ &=\kappa\cdot(\tilde{\mathbf{H}}_{l-1}-\mathbf{H}_{l-1})+\kappa\cdot(1-a)\cdot\mathbf{H}_{l-1}-\kappa\cdot\boldsymbol{\omega}_{l-1}+(1-\kappa)\cdot(\sqrt{\upsilon})^{-1}\cdot\mathbf{V}_l\\ &=\kappa\cdot\mathbf{e}_{l-1}-\kappa\cdot\boldsymbol{\omega}_{l-1}+(1-\kappa)\cdot(\sqrt{\upsilon})^{-1}\cdot\mathbf{V}_l,\end{aligned}\tag{5.60}$$

where it has been considered that $(1–a) \approx 0$. The parameter υ stands for the linear value of the SNR gain obtained from the frequency direction filter.

Taking the result in (5.60), the autocorrelation matrix of the tap error can be expressed as follows,

$$\mathbf{R}_{\mathbf{e}}(l)=\kappa^2\cdot\mathbf{R}_{\mathbf{e}}(l-1)+\kappa^2\cdot\sigma_\omega^2\cdot\mathbf{I}+(1-\kappa)^2\cdot\upsilon^{-1}\cdot\sigma_{\mathrm{V}}^2\cdot\mathbf{I}, \tag{5.61}$$

where **I** stands for the N_u-by-N_u identity matrix and $\sigma_V{}^2$ is the variance of the channel noise after DFT calculation. In the steady state, i.e. $l \rightarrow \infty$, the autocorrelation matrix results as follows

$$\mathbf{R}_{\mathbf{e}}(\infty)=\left(\frac{\kappa^2}{1-\kappa^2}\cdot\sigma_\omega^2+\frac{1-\kappa}{1+\kappa}\cdot\frac{\sigma_{\mathrm{V}}^2}{\upsilon}\right)\cdot\mathbf{I}, \tag{5.62}$$

from where it is straightforward to see that $|\kappa| < 1$ for convergence.

The MSE is obtained as the trace of matrix $\mathbf{R}_\mathbf{e}(\infty)$ (summation of the elements of the main diagonal), yielding

$$\mathrm{MSE}(\kappa)=\mathrm{N}_{\mathrm{u}}\cdot\frac{\sigma_{\mathrm{V}}^2}{\upsilon}\left(\frac{\kappa^2}{1-\kappa^2}\cdot\eta^2+\frac{1-\kappa}{1+\kappa}\right), \tag{5.63}$$

where

$$\eta=\sqrt{\sigma_\omega^2\cdot\frac{\upsilon}{\sigma_{\mathrm{V}}^2}} \tag{5.64}$$

is defined as a *degree of non-stationarity* [Hay02].

The value of MSE given in (5.63) is a convex function with respect to κ and is represented in Figure 5.12 (normalized with respect to $N_u\cdot(\sigma_V{}^2/\upsilon)$) for different values of η. By deriving (5.63) with respect to κ one may obtain the optimal value of κ that minimizes the MSE, yielding

$$\kappa_{opt}=1+\frac{\eta^2}{2}-\sqrt{\eta^2+\frac{\eta^4}{4}}\,. \tag{5.65}$$

It can be seen from Figure 5.12 that for a fixed SNR and symbol length, if the mobile speed (or Doppler bandwidth) is increased, η^2 increases and the MSE shows a very flat behavior (see Figure 5.12a). In this case, a large range of values for κ can be taken, which will provide a relatively constant value of MSE. On the opposite case, when the mobile speed is very small, Figure 5.12c shows that the best solution is to have $\kappa \approx 1$, i.e. updates of the channel estimation are no longer necessary. This is going to be the common scenario found in a WLAN system. Nevertheless, the time direction filter described in this section represents a first order IIR filter used to track the channel CTF. Other more complex tracking algorithms, e.g. Kalman filters, may be used as time direction filters at the expense of an increased computational complexity [Hay02].

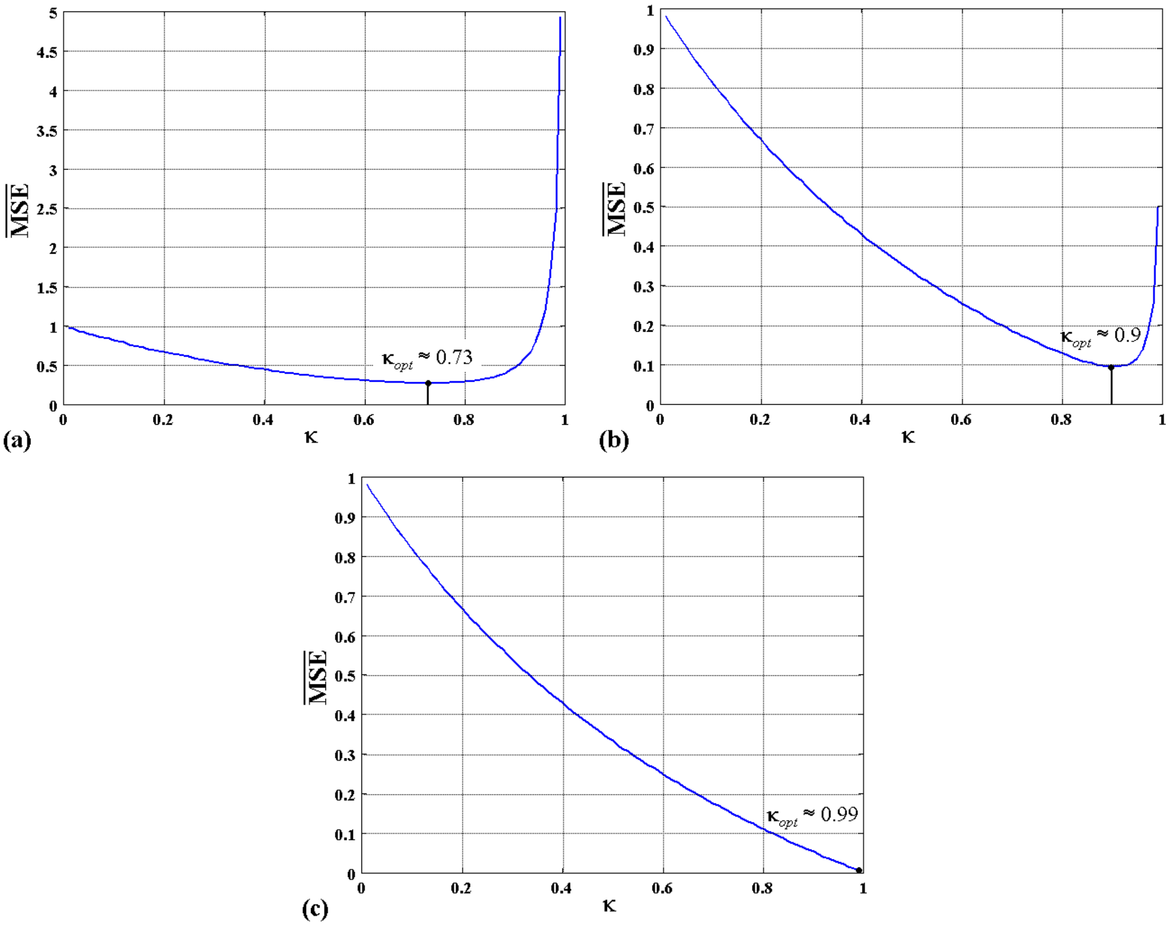


Figure 5.12. Normalized MSE for the time direction filter with: a) $\eta^2 = 10^{-1}$; b) $\eta^2 = 10^{-2}$; c) $\eta^2 = 10^{-4}$.

5.3.4 Implementation Issues

The decision-directed mechanism depicted in Figure 5.11 shows a number of issues to be considered during implementation. Firstly, it considers that the feedback loop introduces a delay of one symbol. In a real implementation the feedback delay depends on the selected traceback length in the Viterbi decoder. In the literature it is mentioned that if no puncturing is used the minimum length should be 5·K [Pro95], where K is the length of the convolutional encoder. If some kind of puncturing is used, it is recommended to use a traceback length in the range 7·K to 9·K or even higher [Flem03]. In order to be able to make a symbol-by-symbol decoding, the authors in [Mig96] propose to drive the convolutional encoder on the transmitter side to the *zero state* at the end of each OFDM symbol. Nevertheless this solution reduces the payload available at each symbol and therefore it is not applicable when the symbol contains a small number of sub-carriers.

Another important point to be considered is the reliability of the channel estimation. In case a very big attenuation is present in some of the sub-carriers, the Equalizer will greatly amplify the noise at those particular sub-carriers. In this case two strategies are possible. A first proposal consists on introducing a gain limit when performing the LS channel estimations. According to [Cim85] there exists an optimum value for the gain limit that depends on the SNR and the expected signal-to-distortion ratio. A second possibility is the inclusion of a reliability estimator. In case a channel coefficient is found to be below a certain threshold, the least reliability value may be assigned to that sub-carrier during soft demapping (see §7.6.3).

5.3.5 Performance Evaluation of the Decision-Directed Channel Estimator

In a decision-directed channel estimator like the one in Figure 5.11 it is of major importance to know how the Traceback Length (TBL) in the Viterbi decoder affects the performance. Initially, this performance is measured in terms of Error Vector Magnitude (EVM) as defined in (7.32). Simulation results are shown in Figures 5.13-5.20 for all the possible modulation schemes found in the IEEE 802.11a standard. In all these cases the EVM is calculated in an AWGN scenario for five different values of TBL: 30, 40, 50, 60 and 120 bits.

Two main results can be derived from Figures 5.13-5.20. On one hand, Figures 5.18 and 5.20 show that a TBL of at least 50 bits should be selected in the Viterbi decoder. These figures correspond to the 36 and 54 Mbps transmission rates, which are particularly sensitive due to the modulation scheme (16-QAM, 64-QAM) and coding rate (¾) they use. On the other side, simulation results show that in all the cases the EVM is very much degraded until a certain value of SNR is reached at the input. This SNR threshold, SNR_T, depends on the particular modulation scheme, coding rate and TBL being used. For values of SNR higher than SNR_T, the EVM is reduced linearly (on a logarithmic scale) as the SNR increases.

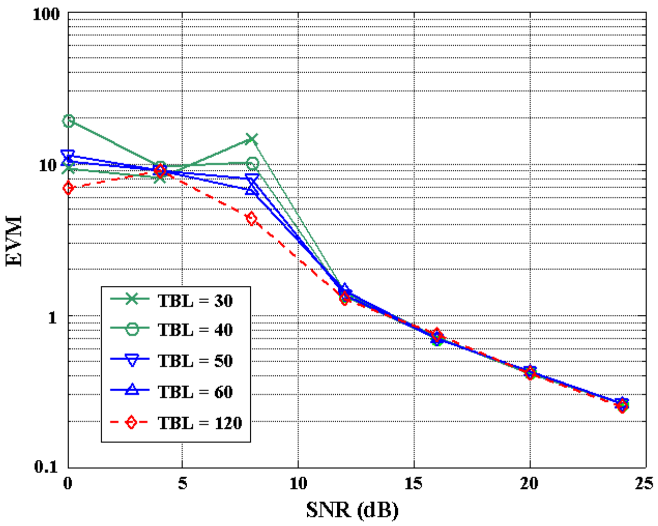


Figure 5.13. Variation of the EVM versus SNR for different traceback lengths (6 Mbps case, AWGN).

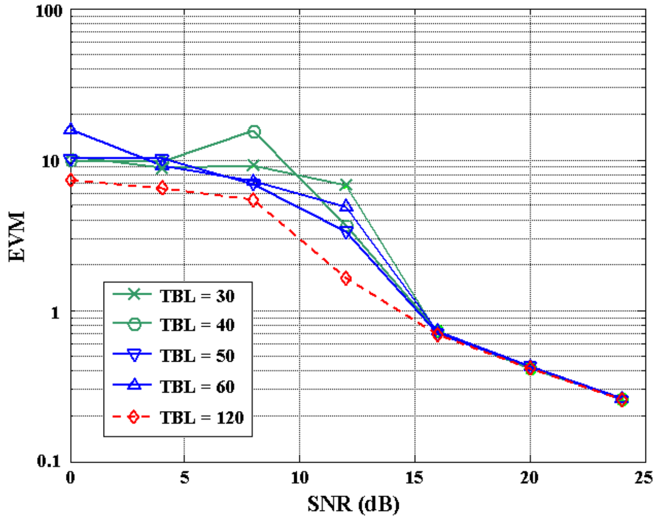


Figure 5.14. Variation of the EVM versus SNR for different traceback lengths (9 Mbps case, AWGN).

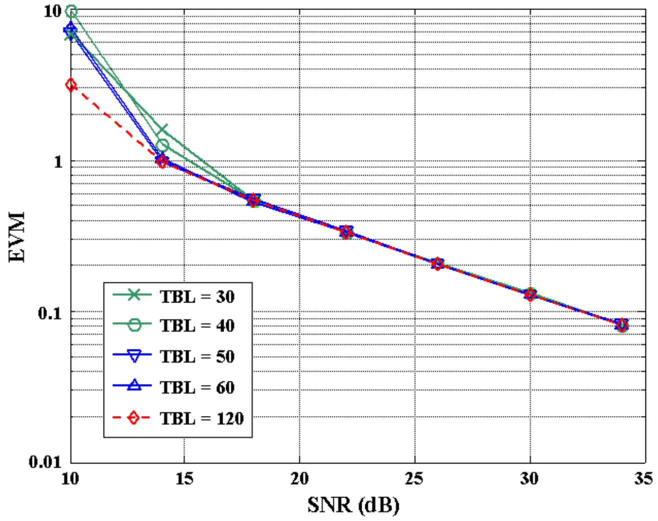


Figure 5.15. Variation of the EVM versus SNR for different traceback lengths (12 Mbps case, AWGN).

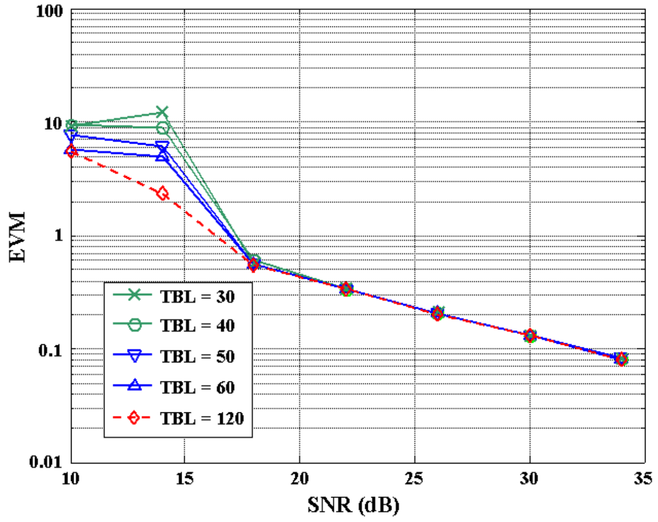


Figure 5.16. Variation of the EVM versus SNR for different traceback lengths (18 Mbps case, AWGN).

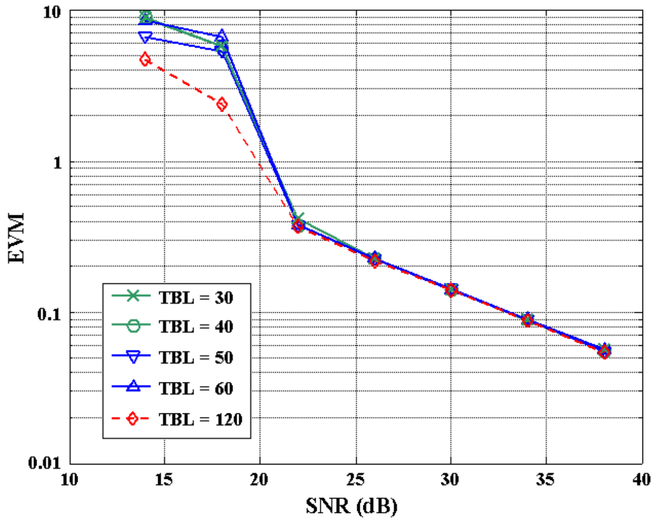


Figure 5.17. Variation of the EVM versus SNR for different traceback lengths (24 Mbps case, AWGN).

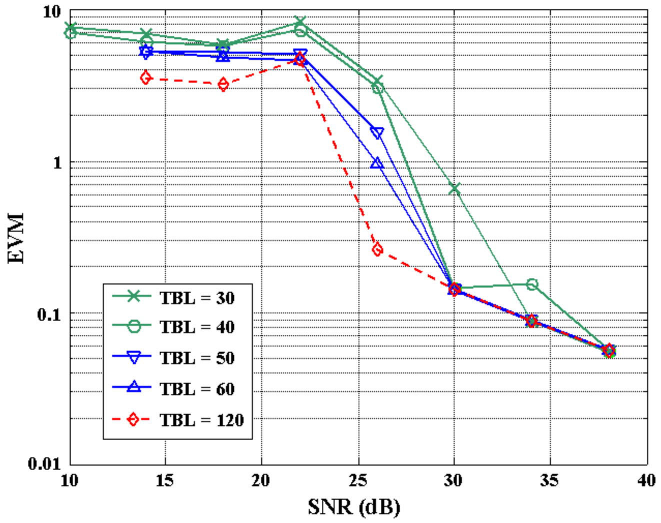


Figure 5.18. Variation of the EVM versus SNR for different traceback lengths (36 Mbps case, AWGN).

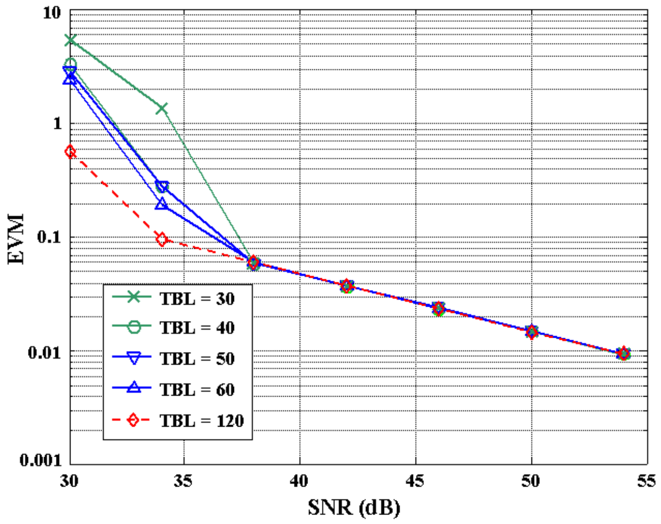


Figure 5.19. Variation of the EVM versus SNR for different traceback lengths (48 Mbps case, AWGN).

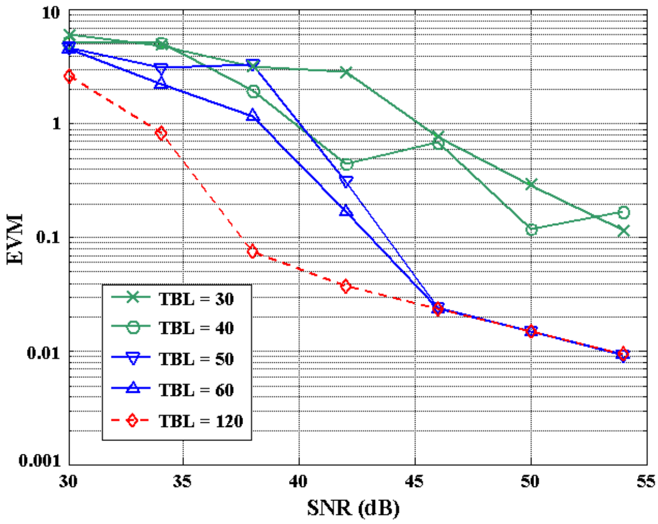


Figure 5.20. Variation of the EVM versus SNR for different traceback lengths (54 Mbps case, AWGN).

The Figures 5.21-5.24 show the variation of the MSE with respect to the SNR for all the modulation schemes in the IEEE 802.11a standard. The channel models A and D were considered in the simulations as given in Table D.1 and D.4, respectively. For reasons that will be explained in §7.3, the parameter κ in the time direction filter was set to 0.001. The TBL in the Viterbi decoder was fixed to 50 bits considering the previous results shown in Figures 5.12-5.20. These Figures show a big improvement in the MSE performance when a Line-Of-Sight is present (Channel D). This is specially the case for those modulation schemes in which puncturing is used. Therefore, we can conclude that the modulation schemes corresponding to the 9, 18, 36, 48 and 54 Mbps modes can only be used under LOS conditions.

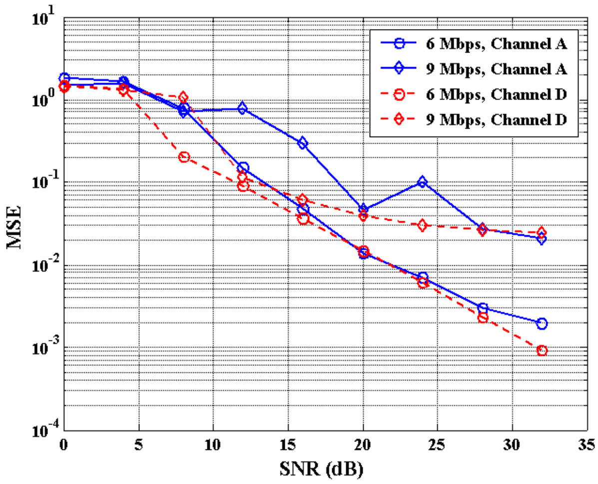


Figure 5.21. MSE versus SNR for the DD channel estimator (BPSK modulation schemes according to the 802.11a standard).

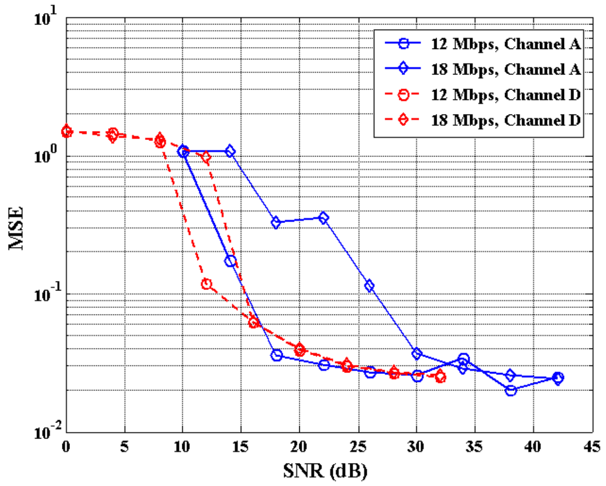


Figure 5.22. MSE versus SNR for the DD channel estimator (QPSK modulation schemes according to the 802.11a standard).

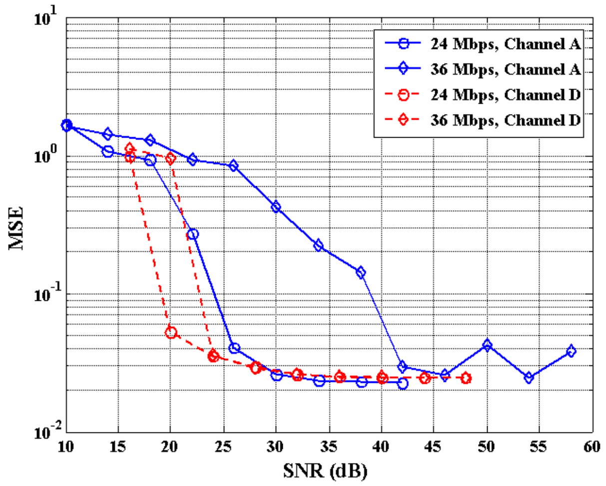


Figure 5.23. MSE versus SNR for the DD channel estimator (16-QAM modulation schemes according to the 802.11a standard).

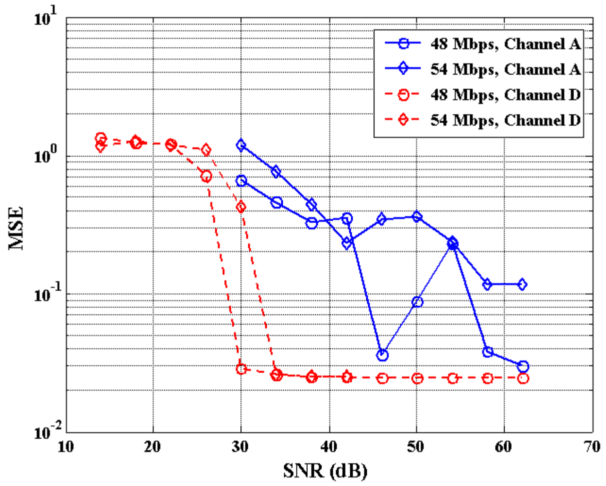


Figure 5.24. MSE versus SNR for the DD channel estimator (64-QAM modulation schemes according to the 802.11a standard).

5.4 Summary

In the present Chapter, Data-Aided (DA) and Decision-Directed (DD) channel estimation techniques applied to OFDM have been presented. The DA techniques are based on the interpolation of pilot tones. Several methods to carry out this interpolation have been analyzed namely, 2-D Wiener, 2x1-D Wiener, 2-D Sinc, 2x1-D Sinc and Low-Rank. The separation of the two-dimensional (2-D) interpolation into two one-dimensional (2x1-D) filters turns out to provide an improvement in the MSE, as the edge effects have a lower impact in this case.

Nevertheless, since the use of pilots reduces the efficiency of the transmission scheme, a DD method was also investigated in which no pilot tones are required. Instead, *virtual pilots* are generated based on the demodulated/decoded information. In order to reduce the noise induced through the feedback loop, two different filters are proposed. The first one is applied in the frequency direction and is based on the low-rank method. The second one is a time direction filter that tries to track the variations in the wireless channel. Since the time direction filter only consists of a 1st-order IIR filter, its tracking performance is very poor. For more accurate tracking, other solutions might be considered. The low-rank principle could be applied in the time direction as it was done in the frequency direction. In this case however, latency is the major issue since the filtering cannot be realized until a certain number of symbols have been collected. On the other side, Kalman filters are seen to provide very good tracking performance [Hay02] but at the expense of higher complexity. Yet another possibility is to use Wiener interpolation filters as frequency and time direction filters, as suggested by Brühninghaus in [Brue99]. Due to the lack of pilot tones, the covariance matrix has to be recalculated on a symbol basis according to the decoded information. The whole solution is computationally very intensive since a matrix inversion needs to be calculated at each symbol time.

A number of simulations were carried out in order to analyze the performance of the proposed DD scheme. Firstly, the EVM was measured for all the modulation schemes defined in the IEEE 802.11a standard using different values of Traceback Length in an AWGN scenario. A Traceback Length of 50 bits was found to be the minimum requirement in the most sensitive transmission rates. The MSE performance was measured afterwards for all the modulation schemes considering a TBL of 50 bits in a wireless scenario (channel models A and D). Results show that those modulation schemes with a code rate different of 1/2 can only be used in scenarios where LOS exists.

As a concluding remark we should mention that only frequency domain equalization has been considered in this Chapter. Generally a combination of time domain and frequency domain equalization is applied if the CIR is found to be longer than the cyclic prefix extension [Bahai99]. A possible method for CIR truncation is given in [Schmidt00]. This method consists on filtering the OFDM symbols in the time domain through a high-pass filter that is designed based on an initial estimation of the CIR. Unfortunately, this solution not only is computationally complex, but also degrades the SNR, due to the high-pass characteristic of the filter.

Up to this point we have analyzed several solutions for the components conforming the Inner Receiver. The forthcoming Chapters in this Dissertation will adapt the different architectures seen so far to be implemented on a real system.

Chapter 6

Data-Aided Synchronizer Architecture suitable for OFDM-WLAN

6.1 Introduction

OUR AIM IN THIS CHAPTER IS THE DEVELOPMENT of a low-power low-latency synchronizer architecture compatible with the specifications given by the IEEE 802.11a [IEEE] standard. Some of the results obtained in the next lines have been published by the author in [Troya02] and [Krstic03a]. The implementation is a derivation of the different Data-Aided solutions investigated in Chapter 4.

The baseband processing of both standards was earlier introduced in §2.4. As depicted in Figure 2.20, a number of *preamble* symbols are appended at the beginning of each frame before transmission. During the reception of that frame, the following operations have to be carried out:

1) Frame detection.
2) Carrier frequency offset estimation and correction.
3) Determination of the symbol timing.
4) Extraction of the reference channel estimation.
5) Data reordering.

The Chapter starts in §6.2 with an explanation of the different preamble structures defined for the standards of interest. All these preambles are modified versions of the preamble structure proposed by Schmidl & Cox in [Schmidl97]. Afterwards, §6.3 describes the architecture of a synchronizer fully compatible with the IEEE 802.11a standard. The proposed design contains novel approaches for several building blocks. Hence, our solution is able to achieve fast frame detection and estimate normalized carrier frequency offsets in the range $|\varepsilon| < 1.5$ by using only two autocorrelators. Actually, only one of those is necessary to perform frame detection by applying a smart plateau detector at its output. In §6.4, a short review of alternative synchronizer architectures proposed in the literature for the 802.11a standard is given with an analysis of their pros and cons. Later on, §6.5 is devoted to the performance evaluation of the proposed synchronizer. Results show that the suggested frame detection algorithm is robust to AGC distortions. In addition, the variance of the normalized frequency estimation error is seen to be below 0.01, therefore complying with the requirements set in §3.2. The section ends with some hints on the application of this solution to the HL2 standard. The present Chapter is finalized with a general summary of results in §6.6.

6.2 Structure of the Preamble Symbols

6.2.1 The IEEE 802.11a Preamble

The 802.11a standard specifies one single preamble structure as depicted in Figure 6.1. It is compound of four OFDM symbols. The first two symbols, i.e. *short preamble* symbols, contain 10 repetitions of the same sequence t_P, each being 16 samples long. The short preamble symbols are generated through the following normalized sequence in the frequency domain:

$$S_{-26,+26} = \sqrt{(13/6)} \cdot \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, \mathbf{0}, \\ 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\}, \qquad (6.1)$$

with $j = \sqrt{(-1)}$. The bold **0** indicates the DC sub-channel.

Note that the samples corresponding to sub-carriers –32 to –27 and +27 to +31 are set to zero (virtual sub-carriers). In order to obtain an arrangement as in Figure 6.1 for the short preamble symbols, the sequence in (6.1) has to be reorganized prior to IDFT calculation. Therefore, the samples are inserted into the IDFT block in the following order including the virtual sub-carriers: 0, +1, +2,…, +31, –32, –31,…, –1. In the sequel, this arrangement of sub-carriers will be referred to as the *natural order*.

The sequence attained after IDFT calculation contains 64 samples and results in the concatenation $\{t_P, t_P, t_P, t_P\}$. This periodic structure is derived from the selected composition of (6.1), where three out of four sub-carriers are set to zero.

The last step consists on the cyclic prefix (CP) insertion. The CP is 16 samples long by definition of the standard, i.e. a sequence equal to t_P is inserted at the beginning in order to generate the concatenated sequence $\{t_P, t_P, t_P, t_P, t_P\}$. Finally, the previous sequence is repeated once so as to obtain the second short preamble symbol, Figure 6.1.

The later two symbols, i.e. *long preamble* symbols, are obtained from the following sequence defined in the frequency domain

$$L_{-26,+26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, \mathbf{0}, \\ 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\}. \qquad (6.2)$$

As it happened with the short preambles, virtual sub-carriers have to be included in the sequence (6.2) and the result is given over to the IDFT block following the natural order. The resulting sequence is the concatenation $\{T_1, T_2\}$ (see Figure 6.1).

The long preamble is obtained after repeating the sequence $\{T_1, T_2\}$ once and appending a cyclic prefix of double size (32 samples / 1.6 μs), equivalent to the sequence T_2. The reason behind that will be explained later in §6.3.5.

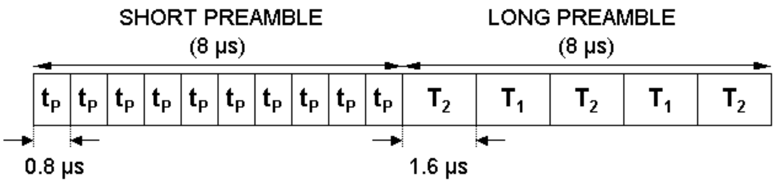


Figure 6.1. Preamble symbols as defined in the IEEE 802.11a standard.

6.2.2 The ETSI BRAN HiperLAN2 Preambles

The preamble definition found in the HL2 standard is somewhat more complicated than in the 802.11a. The several possible preamble structures are depicted in Figure 6.2. Two entities are defined in the wireless network: the Access Point (AP) and the Mobile Terminal (MT). Furthermore, the communication between the AP and the MT is generally referred to as the *downlink*, whereas the communication in the opposite direction (from MT to AP) is called the *uplink*. Depending on whether a certain burst is being transmitted by the former or the later, the corresponding preamble has to be appended.

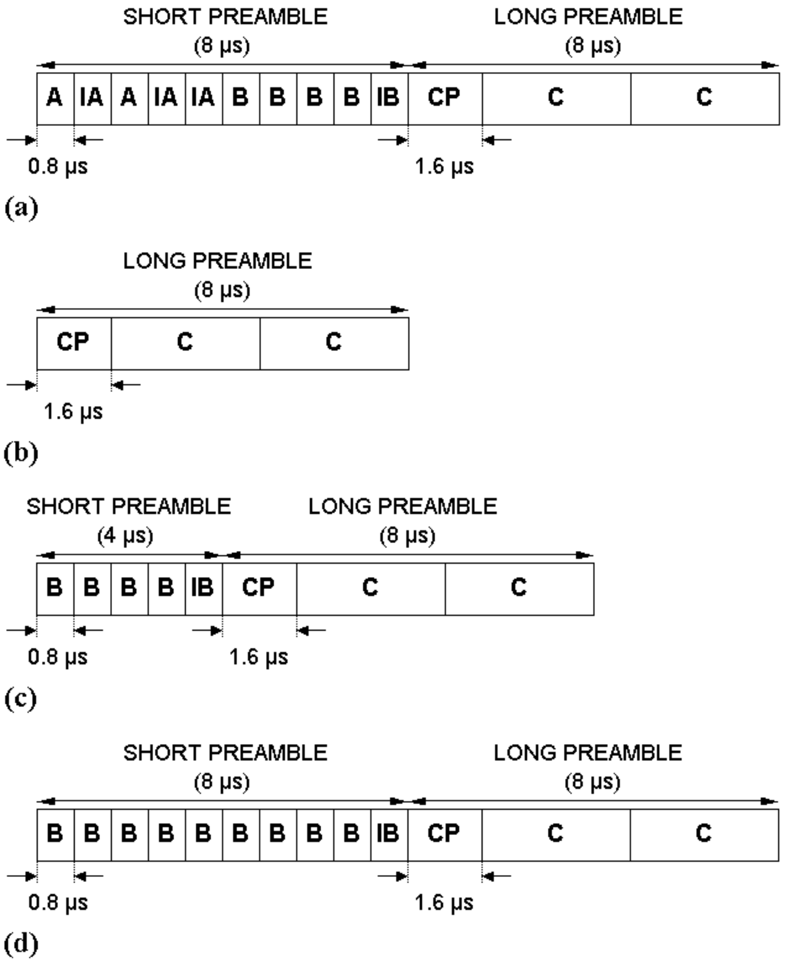


Figure 6.2. Preambles as defined in the ETSI BRAN HiperLAN2 standard: a) Broadcast burst; b) Downlink burst; c) Uplink burst with short preamble; d) Uplink burst with long preamble / Direct link burst.

The preamble shown in Figure 6.2a is the one used in the *broadcast burst*. This burst is always transmitted by the AP in an interval of 2 ms and contains the so-called Broadcast Channel (BCH). The broadcast burst acts as a beacon in which important information related to the network is transmitted. In the other cases, when the AP has to transmit user information to the MT, the preamble used is the one corresponding to the *downlink burst*, as shown in Figure 6.2b.

When the MT is transmitting information towards the AP, either the preamble shown in Figure 6.2c or in Figure 6.2d may be used. The selection depends on which of both is supported by the AP (information provided in the BCH). The preamble shown in Figure 6.2d may also be used in the so-called *direct link bursts*, which occur when two MTs communicate directly without any mediation of the AP and represents an optional feature proposed in the standard.

The generation of these preambles is very similar to the case of the 802.11a standard. Thus, the sequence C in Figure 6.2 is exactly the same as the sequence $\{T_1, T_2\}$ in Figure 6.1 and is obtained from (6.2) using the same procedure. The previous applies also for the concatenated sequence {B, B, B, B} in Figures 6.2a/c/d, which is the same as the sequence $\{t_P, t_P, t_P, t_P\}$, and is generated using (6.1). Nevertheless, the difference is that the last portion of the short preamble symbols is a sign-inverted copy of B, i.e. IB = –B.

The concatenated sequence {A, IA, A, IA} in Figure 6.2a, with IA = –A, is obtained from the following normalized sequence in the frequency domain:

$$SA_{-26,+26} = \sqrt{(13/6)} \cdot \{0, 0, 0, 0, -1+j, 0, 0, 0, 1+j, 0, 0, 0, 1-j, 0, 0, 0, -1-j, 0, 0, 0, -1+j, 0, 0, 0, -1-j, 0, \mathbf{0}, 0, -1+j, 0, 0, 0, -1-j, 0, 0, 0, -1+j, 0, 0, 0, -1-j, 0, 0, 0, 1-j, 0, 0, 0, 1+j, 0, 0, 0, 0\}. \tag{6.3}$$

Again, virtual sub-carriers have to be appended to (6.3) and rearranged according to the natural order before the IDFT is calculated. The final sequence {A, IA, A, IA, IA} is obtained by repeating the last 16 time-domain samples.

6.3 Principle of Operation of the Synchronizer

According to the preamble structures defined in §6.2, an optimum architecture for the synchronizer has to be derived. Obviously, the design will be very much based on the results presented in §4.3 when dealing with Data-Aided synchronization algorithms. The following operations have to be carried out:

1. *Frame detection*. On its default mode, the synchronizer will peer the channel in order to detect an incoming frame. The biggest challenge at this point is the noise amplification made by the Automatic Gain Control (AGC) when no frame is present. The purpose of the AGC is to keep a constant power at its output expanding the signal to the maximum input range of the Analog-to-Digital Converters (ADC). The frame detection algorithm should be robust enough to avoid false alarms caused by the high noise levels.
2. *Timing estimation*. Once the frame is detected, the synchronizer should provide the correct symbol timing in order to differentiate preamble symbols from SIGNAL and DATA symbols, and be able to extract the CP.
3. *Carrier frequency estimation*. The RF oscillators are not exactly tuned at the expected carrier frequency. The remaining frequency error leads to an unacceptable loss of orthogonality and

consequent ICI noise, preventing a correct data demodulation. The synchronizer should perform the estimation and the correction of the frequency offset.

4. *Channel reference estimation*. The last operation expected from the synchronizer is the initialization of the channel estimation/equalization algorithm providing a reference estimation of the CTF.

A flowchart representing all the operations involved in the synchronizer is given in Figure 6.3. In this section, a particular solution for the 802.11a standard will be analyzed, which can be partly used for the HiperLAN2 standard.

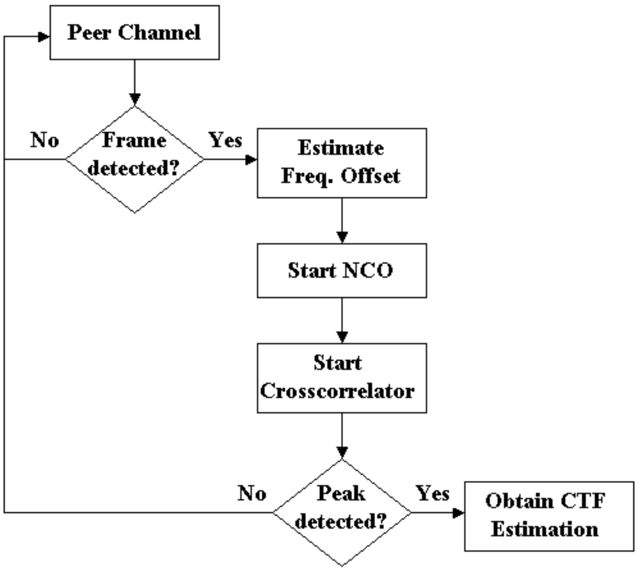


Figure 6.3. Flowchart with the different operations carried out at the Synchronizer.

6.3.1 The Autocorrelator

The periodic structure forming the short preamble symbols in Figure 6.1 suggests a solution similar to the one presented in §4.3.2 for the coarse timing estimation (or frame detection): the autocorrelator. As it was shown there (Figure 4.9) this solution may also be used to perform carrier frequency offset estimations. A more general scheme of the autocorrelator is depicted in Figure 6.4. There, two main parameters are identified: N_d and N_{avg}. The former relates to the amount of delay introduced to the signal whereas, the later indicates the length of the moving average filter.

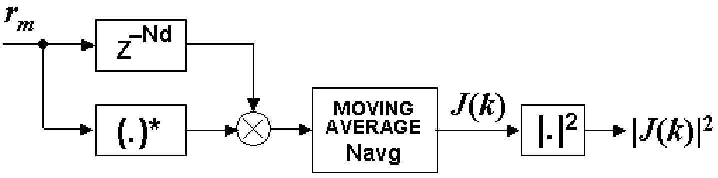


Figure 6.4. General block diagram of the autocorrelator.

Let's consider that the input signal can be expressed as follows,

$$r(m) = s(m) \cdot e^{j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot m} + v(m)\,, \tag{6.4}$$

where $s(m)$ is the original OFDM transmitted signal, ε stands for the normalized carrier frequency offset, Δf is the sub-carrier spacing in the OFDM signal and f_S is the sampling frequency. The sequence $v(m)$ represents a zero-mean white Gaussian noise process. In the case of the IEEE 802.11a standard, Δf = 312.5 kHz and f_S = 20 MHz.

The signal $J(k)$ in Figure 6.4 may be written as

$$\begin{aligned}
J(k) = &\sum_{l=0}^{\mathrm{N_{avg}}-1} r^*(l-k)\cdot r(l-k-\mathrm{N_d}) \\
= &\sum_{l=0}^{\mathrm{N_{avg}}-1}\left(s(l-k)\cdot e^{j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot(l-k)} + v(l-k)\right)^* \cdot \left(s(l-k-\mathrm{N_d})\cdot e^{j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot(l-k-\mathrm{N_d})} + v(l-k-\mathrm{N_d})\right) \\
= &\, e^{-j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot\mathrm{N_d}} \cdot \left\{ \sum_{l=0}^{\mathrm{N_{avg}}-1} s^*(l-k)\cdot s(l-k-\mathrm{N_d}) \right. \\
&+ \sum_{l=0}^{\mathrm{N_{avg}}-1}\left(s^*(l-k)\cdot v(l-k-\mathrm{N_d})\cdot e^{-j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot(l-k)} + v^*(l-k)\cdot s(l-k-\mathrm{N_d})\cdot e^{j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot(l-k-\mathrm{N_d})}\right) \\
&\left. + \sum_{l=0}^{\mathrm{N_{avg}}-1} v^*(l-k)\cdot v(l-k-\mathrm{N_d}) \right\}.
\end{aligned} \tag{6.5}$$

By considering the sequence $s(m)$ to be uncorrelated with the noise $v(m)$, the last two summands in (6.5) can be neglected for sufficiently large values of $\mathrm{N_{avg}}$, yielding

$$\begin{aligned}
J(k) &\approx e^{-j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot\mathrm{N_d}} \cdot \sum_{l=0}^{\mathrm{N_{avg}}-1} s^*(l-k)\cdot s(l-k-\mathrm{N_d}) \\
&= e^{-j2\pi\cdot\varepsilon\cdot\frac{\Delta f}{f_\mathrm{S}}\cdot\mathrm{N_d}} \cdot \sum_{l=0}^{\mathrm{N_{avg}}-1} |\, s(l-k)\,|^2\,,
\end{aligned} \tag{6.6}$$

where it has been considered that the signal $s(m)$ is periodic with a period of $\mathrm{N_d}$ samples, i.e. $s(m) = s(m-\mathrm{N_d})$. From (6.6) it is straightforward to see that the phase of $J(k)$ is only due to ε, and so ε could be found as follows

$$\varepsilon = \frac{f_\mathrm{S}}{2\pi\cdot\mathrm{N_d}\cdot\Delta f}\tan^{-1}(J^*(k))\,. \tag{6.7}$$

However, there are several factors which destroy the periodicity, making $s(m) \neq s(m-\mathrm{N_d})$. The most important ones are the AGC settling time and the CIR. The noise also contributes, but its effect can be largely compensated by the averaging $\mathrm{N_{avg}}$ in (6.5). In addition, if $\mathrm{N_{avg}}$ is a multiple of the minimum periodicity in the preambles (16 samples in the standard), the signal $|J(k)|^2$ in Figure 6.4 shows a plateau in the region where the phase of $J(k)$ only depends on the carrier frequency offset. This plateau is shown in Figure 6.5, when the autocorrelator in Figure 6.4 is applied to the 802.11a preamble with different values for $\mathrm{N_d}$ and $\mathrm{N_{avg}}$.

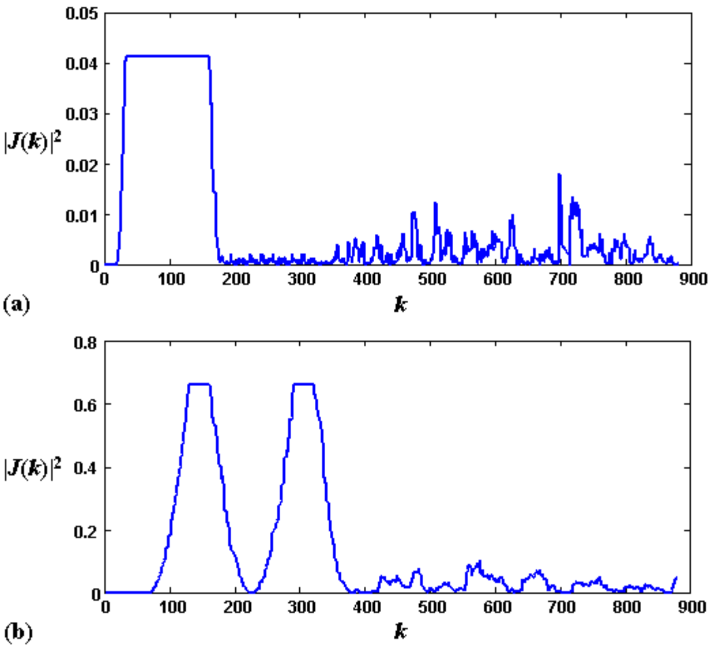


Figure 6.5. Autocorrelation applied to the IEEE 802.11a preambles: a) N_d=16, N_{avg}=16; b) N_d=64, N_{avg}=64.

The Arctangent operation found in equation (6.7) is restricted to the range [–π, π). Hence, there is a limitation in the frequency estimation given by

$$-\pi \le 2\pi \cdot \varepsilon \cdot \mathrm{N}_d \cdot \frac{\Delta f}{f_S} < \pi , \tag{6.8}$$

or equivalently

$$|\varepsilon| < \frac{f_S}{2 \cdot \mathrm{N}_d \cdot \Delta f} . \tag{6.9}$$

Since the ratio (f_S / Δf) is a fixed parameter, the range of possible estimated values for ε will only depend on the selected delay N_d in the autocorrelator. In the standards under consideration, the following parameters are taken: f_S = 20 MHz and Δf = 312.5 kHz (no oversampling is considered at the synchronizer), resulting in $|\varepsilon| <$ 0.5 for N_d = 64 and $|\varepsilon| <$ 2.0 for N_d = 16. The estimation of ε can be obtained from any point of $J(k)$ belonging to the plateaus.

6.3.2 Plateau Detection Mechanism

The previous results suggest that the frequency estimation can be realized as soon as the plateau in $|J(k)|^2$ is detected. Hence, a simple mechanism for plateau detection is necessary. For this purpose the scheme depicted in Figure 6.6 is suggested, in which the plateau detector is divided into two sub-blocks: *differentiator* and *peak detector*.

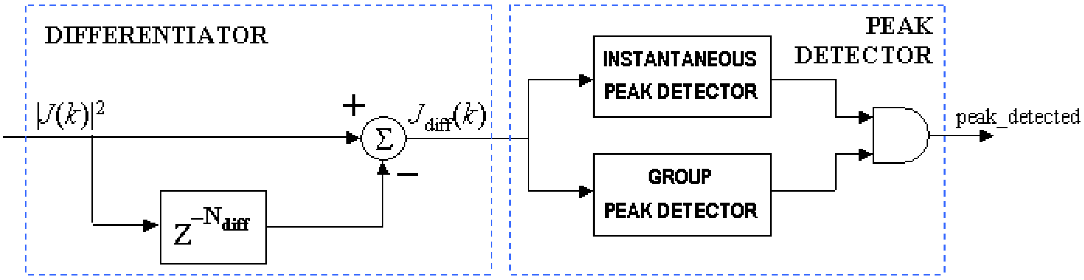


Figure 6.6. General scheme of the proposed plateau detector.

Making use of a differentiator, the algorithm tries to find out the point where the plateau starts. At this point the function $|J(k)|^2$ is not differentiable and an ideal differentiator would show a discontinuity. A real differentiator has a limited bandwidth and so discontinuities may not happen. In addition, in order to obtain a noise-robust system, this bandwidth needs to be as small as possible. The parameter N_{diff} in Figure 6.6 is taken as the duration of the plateau in $|J(k)|^2$. By focusing our attention to the result for $|J(k)|^2$ depicted in Figure 6.5b ($N_d = N_{avg} = 64$), a value of $N_{\mathrm{diff}} = 32$ was selected. For this particular case, the signal $J_{\mathrm{diff}}(k)$ is as shown in Figure 6.7.

An interesting feature in $J_{\mathrm{diff}}(k)$ is that it shows an absolute maximum at the point where the plateau starts (see Figure 6.7). The autocorrelation block together with the differentiator and the peak detector will constantly "peer" the channel. When the peak detector in Figure 6.6 identifies an absolute maximum at the output of the differentiator, the synchronizer will consider that a new frame has arrived and the carrier frequency offset estimator will be activated. Because of the noise, the peak detection will not be a trivial task, i.e. a smart peak detection algorithm will be necessary in order to distinguish absolute from relative maxima. This algorithm is divided into two components, as shown in Figure 6.6: *group peak detector* and *instantaneous peak detector*.

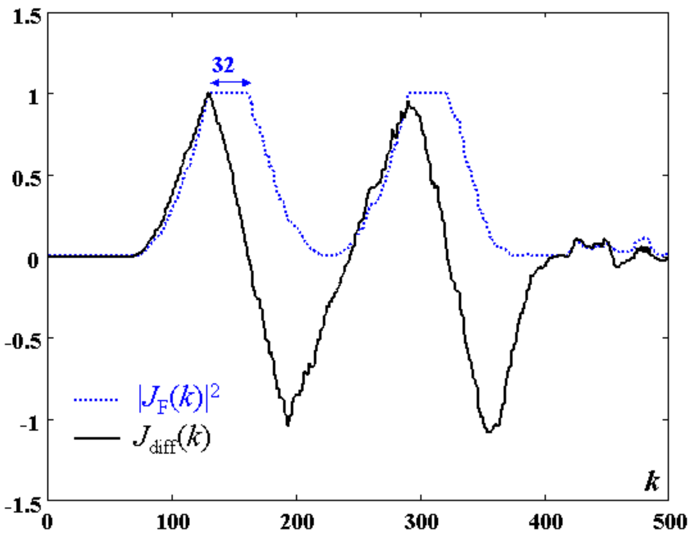


Figure 6.7. Normalized signal at the output of the differentiator block in Figure 6.6.

Figure 6.8 provides a detailed block diagram of the group and instantaneous peak detectors. The **instantaneous peak detector** is basically compound of a comparator and a counter. The present sample

$J_{\mathrm{diff}}(k)$ coming out from the differentiator is compared with the last recorded maximum $J_{\max}$ ($J_{\max} = 0$ at $k = 0$). As long as the sample $J_{\mathrm{diff}}(k)$ is bigger than $J_{\max}$, the register storing $J_{\max}$ will be updated to contain the new sample $J_{\mathrm{diff}}(k)$ as the new encountered maximum and the counter will be reset. If $J_{\mathrm{diff}}(k)$ is smaller or equal than $J_{\max}$, the counter will be triggered and it will increase its count by one. If this situation persists until the counter overflows, the instantaneous peak detector will activate a signal stating that a relative peak was found inside the counting scope of the counter. In the proposed design, the counting range is in the interval $[0, N_{\mathrm{diff}}/2)$. Hence, for our particular implementation, a 4-bit counter is sufficient.

The **group peak detector** is used to detect falling slopes in $J_{\mathrm{diff}}(k)$, and its main block is also a comparator. If the group peak detector finds a falling edge at the same time as the instantaneous peak detector finds a relative peak, this means that the detected peak is actually an absolute peak. In the group peak detector, the input signal is accumulated in groups of N_{acc} samples (Accumulator in Figure 6.8) and the present group is compared with the previous one. If it is not smaller, it means that the falling slope has started.

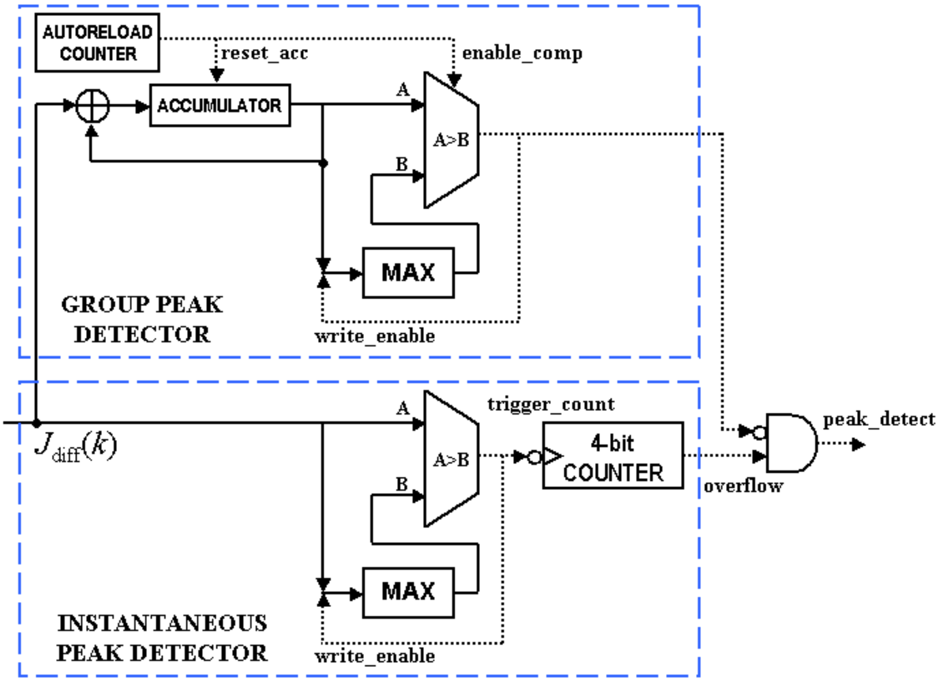


Figure 6.8. Block diagram of the proposed peak detector.

The results of the peak detection algorithm are shown in Figure 6.9 for $N_{\mathrm{acc}} = 6$ (no noise is considered). From Figure 6.9 it is readily to see that the frame will be detected about 16 samples after the starting of the plateau, i.e. in our example the detection will occur in the middle of the first plateau in $|J(k)|^2$. At this point, the carrier frequency offset estimation will be carried out.

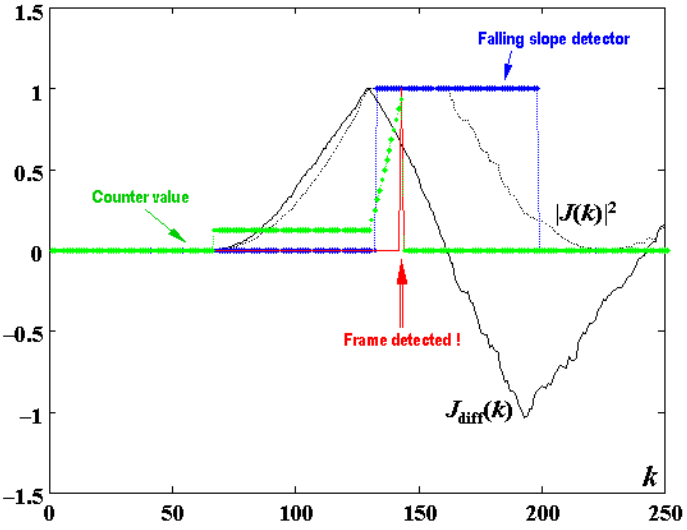


Figure 6.9. Graphical explanation of the frame detection procedure.

6.3.3 Carrier Frequency Offset Estimation and Correction

According to the specifications found for the 802.11a and HL2 standards, all the clocks and carrier signals on the chip should be generated from the same crystal oscillator, which should have a maximum relative frequency error in the range ±20 ppm. Let's consider an example in which a signal is received at the highest possible carrier frequency of 5,805 MHz (operating channel 161 in the U-NII upper band). The total frequency deviation during downconversion is then given as 5,805 · ±20 = ±116.1 kHz. The whole transmit-receive process introduces an overall carrier frequency error of ±232.2 kHz. Normalizing this value with respect to the sub-carrier spacing, $\Delta f = 312.5$ kHz, results in $\varepsilon = \pm 0.75$.

The frequency range that can be estimated using the autocorrelator in Figure 6.4 depends on N_d according to (6.9) (see Table 6.1). As a design constraint, solely values of N_d multiples of 16 can be used, since this is the minimum periodicity found in the preamble. According to Table 6.1, it would be enough to select $N_d = 32$ to achieve the expected range $\varepsilon = \pm 0.75$. However, a cautious design would use two autocorrelators instead of one, with N_d equal to 16 and 32, respectively, and obtain the frequency estimation as an average of both.

The present implementation considers the frequency offsets to be in the range $\varepsilon = \pm 1.5$, i.e. twice the maximum value accepted by the standard. This decision is based on a pessimistic approach and is founded on the fact that functional tests will be carried out using experimental AFE, which most probably will not entirely fulfill the specifications.

Table 6.1. Range of estimation of $|\varepsilon|$ versus N_d.

N_d	$\|\varepsilon\|_{max}$
16	2.0
32	1.0
48	0.66
64	0.5

Two autocorrelators with parameters (N_d = 64, N_{avg} = 64) and (N_d = 16, N_{avg} = 16) were selected in this implementation. The former provides very robust frequency estimation due to the selected high averaging in the range $\alpha \in [-0.5, +0.5)$, which will be referred to as the *fine frequency offset*. The later extends the estimation range to be $\beta \in [-2, 2)$ providing a less robust *coarse frequency offset*. The proposed scheme is depicted in Figure 6.10.

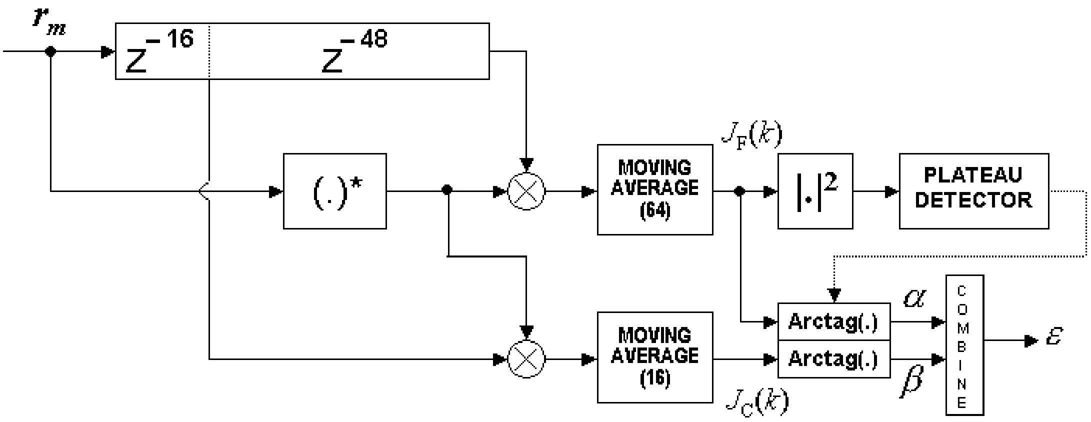


Figure 6.10. Proposed frequency offset estimator.

According to the results provided in §6.3.2, the autocorrelator used to obtain the fine frequency offset may also be used for the coarse timing estimation (frame detection). Thus, a joint timing-frequency estimation is possible. The scheme shown in Figure 6.10 should permanently calculate $J_F(k)$ and $J_C(k)$ (the sub-index states for *Fine* and *Coarse*). Furthermore, $|J_F(k)|^2$ is applied as the input to the plateau detector in Figure 6.6.

Once the detection of a frame is asserted, the two samples $J_F(k)$ and $J_C(k)$ will be stored in a register and the Arctangent calculators will be activated. The Arctangent operation can be efficiently realized by using the CORDIC algorithm working in the *vectoring mode*. Complete details on the theory and realization of the CORDIC algorithm are provided in Appendix A. Although two Arctangent blocks have been drawn in Figure 6.10, only one is necessary. The Arctangent will firstly calculate α from $J_F(k)$ and afterwards β from $J_C(k)$.

The dependency of α and β versus the normalized frequency offset ε is sketched in Figure 6.11. There it can be seen that this dependency is not completely linear, but it shows some discontinuities. Thus, the final value for ε cannot be directly a linear combination of the two estimations α and β.

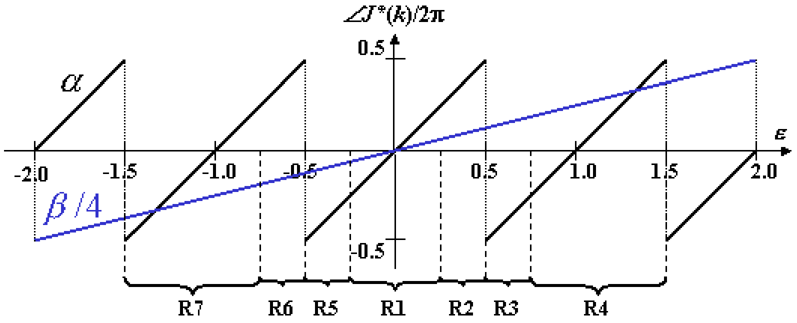


Figure 6.11. Relationship between normalized phase of $J^*(k)$ and frequency offset together with the seven decision regions.

From Figure 6.11 it can also be seen that β has no discontinuities throughout the entire range of expected values for ε (±1.5). Nevertheless, β will be much noisier than α because the moving average considers only 16 samples. For this reason, β will only serve as a range pointer, which will be further refined by α. The final value of ε results from the following function,

$$\begin{array}{lll}
\varepsilon = \alpha & ;\text{ if } -0.25 \leq \beta \leq 0.25 & \text{(R1)} \\
\varepsilon = \alpha & ;\text{ if } \alpha \geq 0 \text{ and } 0.25 < \beta < 0.75 & \text{(R2)} \\
\varepsilon = 1 + \alpha & ;\text{ if } \alpha < 0 \text{ and } 0.25 < \beta < 0.75 & \text{(R3)} \\
\varepsilon = 1 + \alpha & ;\text{ if } \beta \geq 0.75 & \text{(R4)} \\
\varepsilon = \alpha & ;\text{ if } \alpha < 0 \text{ and } -0.75 < \beta < -0.25 & \text{(R5)} \\
\varepsilon = -1 + \alpha & ;\text{ if } \alpha \geq 0 \text{ and } -0.75 < \beta < -0.25 & \text{(R6)} \\
\varepsilon = -1 + \alpha & ;\text{ if } \beta \leq -0.75 & \text{(R7)}
\end{array} \qquad (6.10)$$

The correction of the frequency offset is carried out considering the signal model given in (6.4). In order to obtain the original signal $s(m)$, the input signal $r(m)$ has to be multiplied by a phasor, which is the complex conjugate of the one found in (6.4). This operation will be carried out by a Numerically Controlled Oscillator (NCO), which is implemented using again the CORDIC algorithm, this time in the *rotational mode* (see Appendix A for details on theory and implementation). The NCO will run until the end of the frame is detected. Immediately after frequency correction, the input frame is passed on to the fine timing estimator, which will provide an accurate indication of the symbol timing.

6.3.4 Symbol Timing Estimation

Unlike to what was done during carrier frequency offset estimation, where the periodicity of the short preamble symbols was the main feature exploited by the estimator, the symbol timing estimation will be obtained by exploiting the direct knowledge of the long preamble symbols. Once the frame is detected, the synchronizer knows approximately which samples of the preamble have already entered, but this knowledge is not enough for further processing and has to be refined.

The main block in the symbol timing estimator is a *crosscorrelator*. Its purpose is to compare the input frame with a reference signal, which is directly obtained from the long preamble symbol. The crosscorrelator can only be applied once the samples of the incoming frame have been fully corrected for the frequency offset by the NCO.

The fraction of the long preamble symbol selected as the crosscorrelator reference $c_{\mathrm{REF}}(m)$ is shown in Figure 6.12 and corresponds to the sequence defined as T_1. The reference has a length of 32 complex samples, which is the shortest possible length for this reference in order to obtain appropriate results after correlation. Considering the 802.11a standard, the reference is given as follows,

$c_{\mathrm{REF}}(0{:}31) = \{0.1563,\ -0.0051 - j0.1203,\ 0.0397 - j0.1112,\ 0.0968 + j0.0828,$
$0.0211 + j0.0279,\ 0.0598 - j0.0877,\ -0.1151 - j0.0552,\ -0.0383 - j0.1062,$
$0.0975 - j0.0259,\ 0.0533 + j0.0041,\ 0.0010 - j0.1150,\ -0.1368 - j0.0474,$

$$\begin{gathered}0.0245 - j0.0585,\ 0.0587 - j0.0149,\ -0.0225 + j0.1607,\ 0.1192 - j0.0041,\\ 0.0625 - j0.0625,\ 0.0369 + j0.0983,\ -0.0572 + j0.0393,\ -0.1313 + j0.0652,\\ 0.0822 + j0.0924,\ 0.0696 + j0.0141,\ -0.0603 + j0.0813,\ -0.0565 - j0.0218,\\ -0.0350 - j0.1509,\ -0.1219 - j0.0166,\ -0.1273 - j0.0205,\ 0.0751 - j0.0740,\\ -0.0028 + j0.0538,\ -0.0919 + j0.1151,\ 0.0917 + j0.1059,\ 0.0123 + j0.0976\}, \qquad (6.11)\end{gathered}$$

where $j=\sqrt{-1}$.

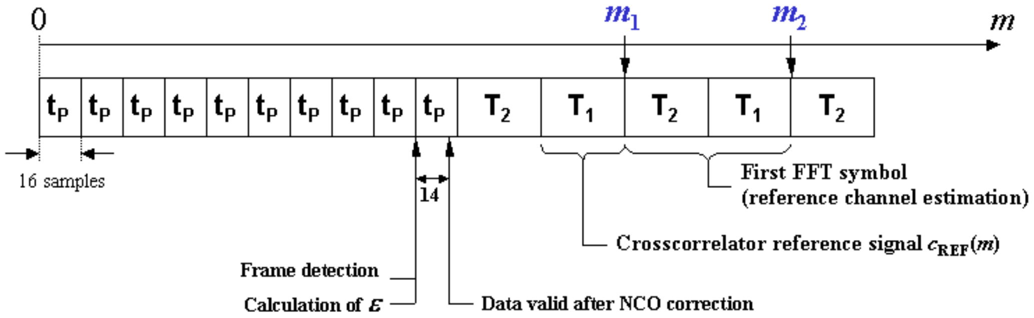


Figure 6.12. Timing scheme for the different operations realized within the synchronizer.

Under an implementation point of view, the complex crosscorrelator is usually a “weak” point in modern communication circuit designs because of its computation complexity, i.e. it requires a large number of complex multipliers, and needs large silicon area. Having this in mind, in this implementation we apply a simplified scheme for the crosscorrelator, with simple XNOR 1-bit multipliers (Figure 6.13) [Troya99] that substitute the commonly used complex multipliers. Instead of multiplying b-bit complex numbers, the XNOR multiplier performs only the multiplication of the sign bits of the complex input values, assigning a '1' when the signal is positive or zero and a '0' when it is negative. In addition, a further simplification in the structure of these multipliers is possible if one of the inputs is fixed and known beforehand. In this simplification, the XNOR gates may be replaced by NOT gates, which require a smaller number of transistors.

The final structure of the crosscorrelator is depicted in Figure 6.14. The structure of each multiplier is decided upon the reference signal $c_{\mathrm{REF}}(31{:}0)^*$, where it has been already considered the reference to be complex conjugated, hard-limited and order-reversed. The final reference looks as follows,

$$c_{\mathrm{REF}}(31{:}0)^* = \{1, 1, 0, 0, 1+j, j, j, j, j, 0, 1, 1, 0, 0, 1, 1+j, 1+j, 0, 1+j, 1+j, j, 1+j, 1, 1+j, j, j, 1+j, 1, 1, 1+j, j, 1\}. \qquad (6.12)$$

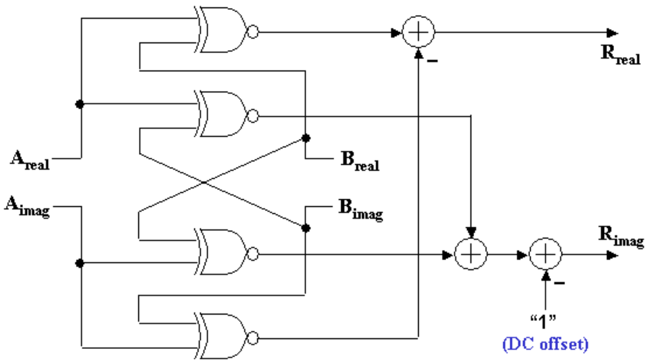


Figure 6.13. Scheme of the XNOR-based complex multiplier.

If the preamble symbols go through the crosscorrelator, the obtained output is as shown in Figure 6.15, where two major peaks become visible at instants m_1 and m_2. Both peaks will happen when the portions T_1 in the long preamble symbols (see Figure 6.12) are inside the crosscorrelator. For our purpose it is enough to detect the first peak by setting a certain threshold at the output of the crosscorrelator. The 64 samples coming immediately after the first peak will be fed into the FFT in order to extract the *reference channel estimation*.

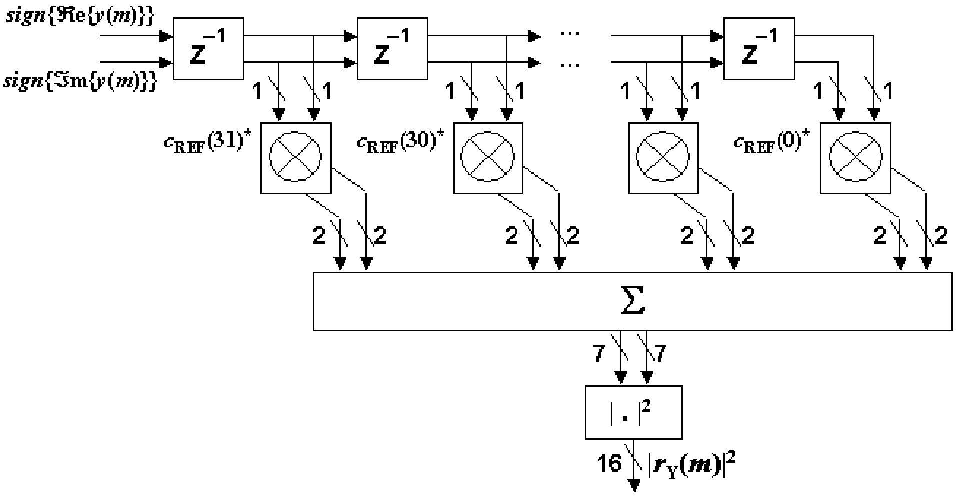


Figure 6.14. Structure of the crosscorrelator considering the simplified XNOR-based architecture.

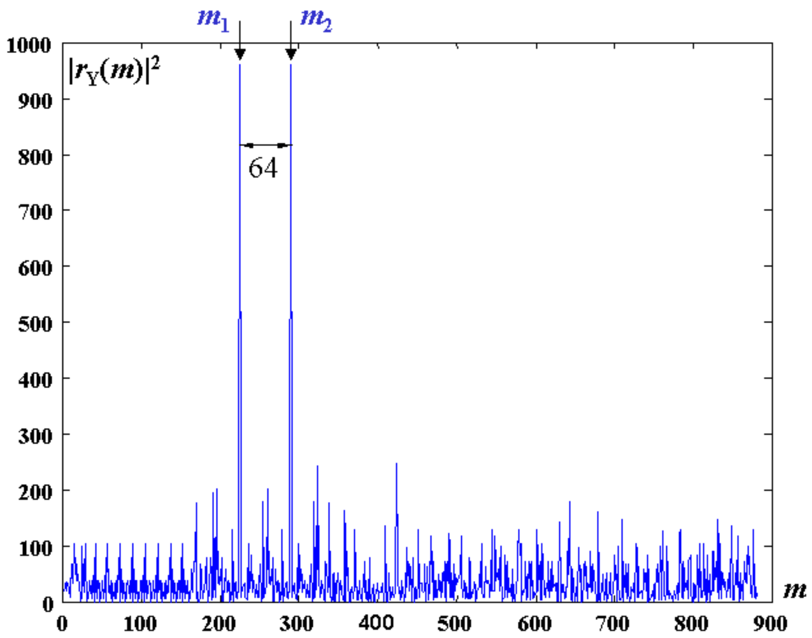


Figure 6.15. Results after crosscorrelation of $c_{REF}(m)$ in (6.12) with the preamble symbols.

6.3.5 Extraction of the Reference CTF

The last operation to be performed by the synchronizer is the extraction of a reference CTF (Channel Transfer Function). The reference CTF is used by the channel estimator as a means for correcting the filtering affecting the OFDM symbols due to the transmission channel and the analog filters found in the AFE. At this point, since the information content found in the long preamble symbols is known beforehand in the frequency domain for all the used sub-carriers, i.e. (6.2), they may be used for the purpose of channel estimation.

The long preamble symbols depicted in Figure 6.1 show a cyclic prefix of double length, i.e. 32 samples or 1.6 μs, plus a double repetition of the sequence $\{T_1, T_2\}$, each being 64 samples long. Each of the two last sequences $\{T_1, T_2\}$ is intended for a single channel estimation. The final reference estimation provided to the *equalization stage* results after averaging the previous two, thus a 3-dB gain in the SNR can be obtained. In addition, the purpose of the double cyclic prefix (initial sequence T_2 in Figure 6.1) is to "absorb" the Inter-Symbol Interference that may result from channels with a CIR longer than 0.8 μs.

In view of that, one may ask why does the standard define the cyclic prefix to be 0.8 μs if longer channels are expected? The answer is that in this case some kind of pre-equalization block prior to FFT calculation should be considered in the implementation in order to shorten the CIR. Nevertheless, such shortening filters are computationally very intensive [Schmidt00] and hard to implement.

Yet a more stringent condition is given by the 802.11a standard regarding the latency in the receive direction. In order to define this latency, let's consider a timer is started at the very moment in which the last sample of an incoming frame reaches the antenna. The frame undergoes the whole digital processing up to the MAC processor. If no error is encountered, the MAC processor generates an ACK (acknowledge) frame to be transmitted. At the moment in which the first sample in the preamble symbol of the ACK frame reaches again the antenna, the timer is stopped. The standard specifies that the time measured by our imaginary timer should be less than 16 μs. Considering the complexity of the whole transmit-receive chain (FFT, Viterbi,…), the previous requirement imposes very though timing conditions on the whole system and specially on the synchronizer. One way to reduce the latency is by avoiding any kind of data buffering during the digital processing.

Recalling the solution given in the previous section for the fine timing estimation, the extraction of the reference CTF should follow just after detection of the peak at position m_1 (see Figures 6.12). This is a necessary condition if buffering is to be avoided. Hence, only a single sequence $\{T_2, T_1\}$ will be available for the reference channel extraction, provoking a degradation of 3 dB in the SNR with respect to the ideal case mentioned above in which two sequences are averaged. The sequence $\{T_2, T_1\}$ represents a cyclically delayed version of the original sequence $\{T_1, T_2\}$ given in the frequency domain by (6.2). After performing the FFT, any time delay is seen as a linear phase, which in this specific case is reduced to the sequence $\exp\{-j2\pi(32/64)k\} = \exp\{-j\pi k\} = (-1)^k$. In order to compensate for this linear phase, the FFT output has to be multiplied exactly by this sequence. Nevertheless, the phase compensation mentioned above is only necessary when calculating the reference CTF. For the data symbols coming after the preamble no phase correction will be done.

There are two further operations to be performed inside the FFT block. The first one occurs before the FFT calculation and relates to the *cyclic prefix extraction*. For each data symbol, 80 samples are expected. The first 16 will be discarded and the last 64 will be fed into the FFT. The second operation takes place after FFT and concerns the *channel reordering*. After FFT operation, the data is delivered following the natural order as mentioned in §6.2.1, i.e. the sequence of 64 samples is considered to be in the following order including the virtual sub-carriers: 0, +1, +2,…, +31, −32, −31,…, −1. For further processing, this order has to be changed and the virtual as well as the DC sub-carriers should to be discarded. Furthermore, the pilot sub-carriers will be sign-corrected according to the standard (see §2.4.1) and delivered in first place before the data sub-carriers. The resulting order is as follows: −21, −7, +7, +21, −26, −25,…, −22, −20, −19, −18,…, −8, −6, −5,…, −1, +1,…, +5, +6, +8,…, +18, +19, +20, +22,…, +25, +26. The data re-ordering significantly contributes to the overall latency since it requires a partial data buffering. The reference CTF will be delivered to the channel estimator following the previous order as well. In addition, the obtained channel reference has to be sign-corrected since it is BPSK modulated in the frequency domain as given in (6.2). The further processing of the data symbols just requires the frequency correction through the NCO and the FFT operation until the end of the frame is detected. Details on the VLSI implementation of the proposed synchronizer architecture are addressed in Chapter 8.

Other approaches to the design of the synchronizer may be considered. Next section deals with two particular solutions, one proposed by a research group at the Berkeley Wireless Research Center and the other one by Nokia. Both solutions are computationally more intensive than the proposed one, specially in the initial phase when the channel is peered.

6.4 Alternative Architectures

Alternative solutions for the synchronizer are proposed in [Chiu00] and [Schwoe01], especially with respect to the frame detection mechanism. The solution given in [Chiu00] is depicted in Figure 6.16. It can be seen that the carrier frequency estimation is as proposed in §6.3.3. Nevertheless, the way α and β are combined is not fully clarified by the authors. The biggest difference lies in the method used for frame detection. Two signals play a key role in this scheme as becomes visible from Figure 6.16: $|J_C(k)|^2$ and $|R(k)|^2$.

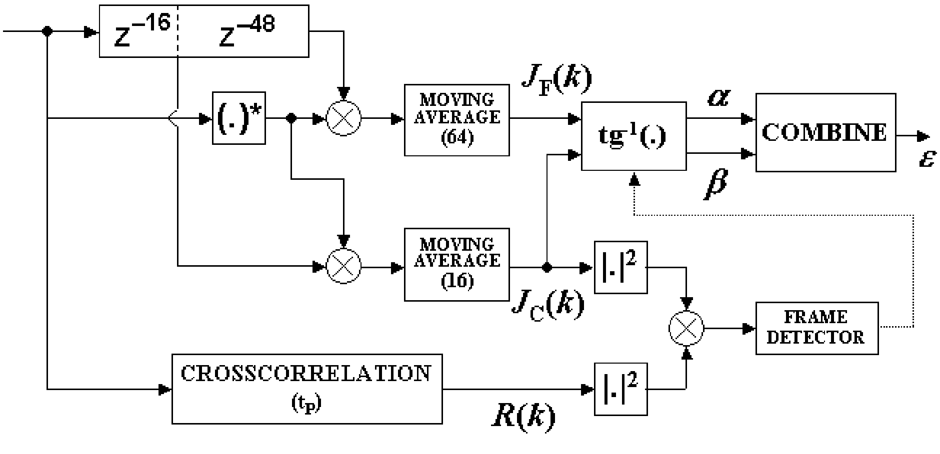


Figure 6.16. Alternative #1 for the synchronizer according to [Chiu00].

The signal $|J_C(k)|^2$ shows a very long plateau of 128 samples (see Figure 6.5a). The signal $R(k)$ is the result of the crosscorrelation between the input frame and the sequence t_P, where t_P corresponds to a short preamble symbol (see Figure 6.1). The result of this crosscorrelation is shown in Figure 6.17a for the ideal case (i.e. no noise, no carrier frequency offset and no fading channel). The result of the multiplication between $|J_C(k)|^2$ and $|R(k)|^2$ is depicted in Figure 6.17b.

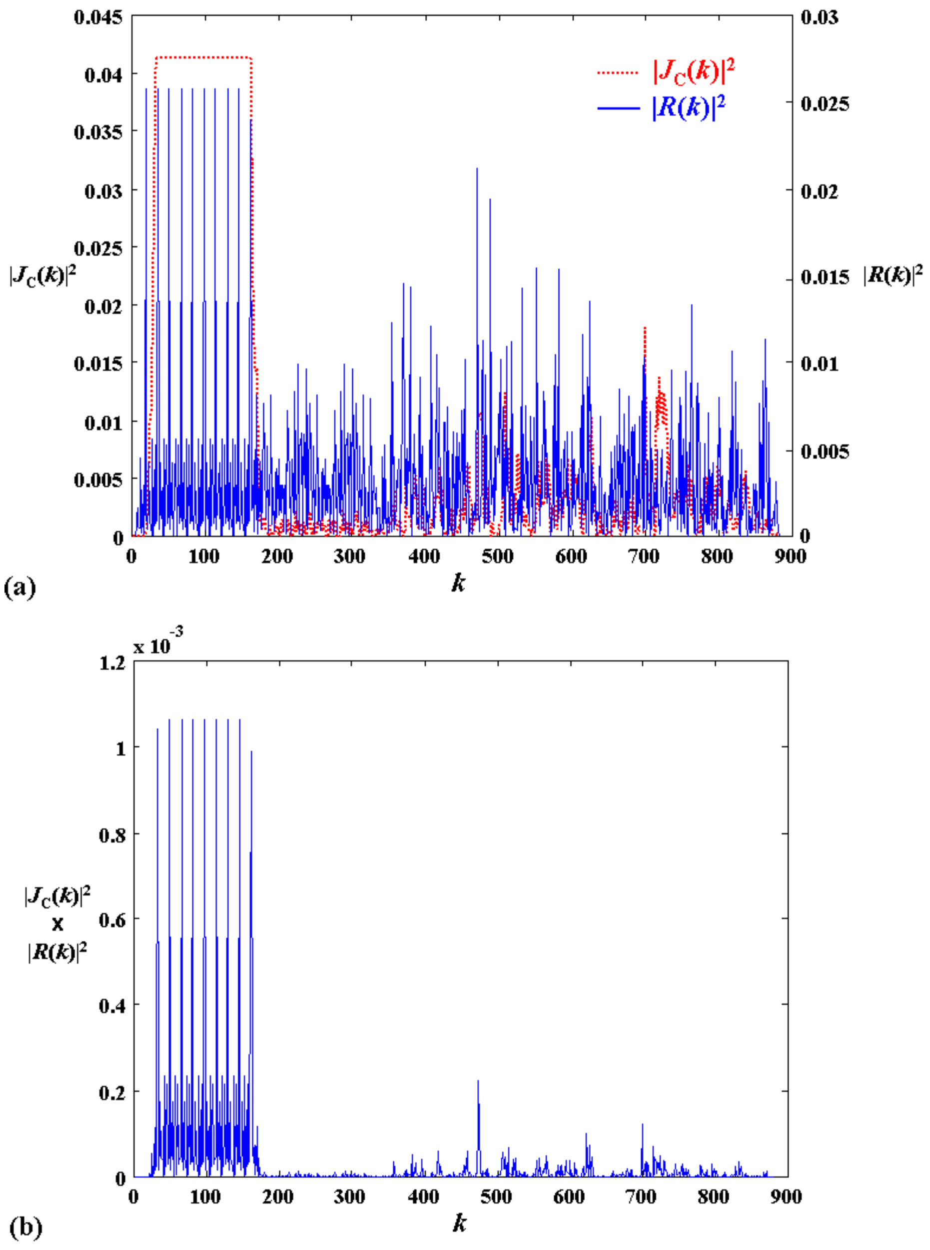


Figure 6.17. Signals involved in the frame detection algorithm proposed in [Chiu00]: a) Representation of $|J_C(k)|^2$ and $|R(k)|^2$; b) Product of $|J_C(k)|^2$ and $|R(k)|^2$.

The multiplied signal depicted in Figure 6.17b shows a number of clearly defined peaks that can be used to position the synchronizer inside the frame. Nevertheless, the simplified crosscorrelator given in §6.3.4 cannot be used in this case for two main reasons: i) the reference signal is too short (only 16 samples); ii) the input signal is affected by a frequency offset. Under these conditions the simplified crosscorrelator does not perform adequately. Hence, a full crosscorrelator needs to be implemented, which is going to be active as long as the channel is being sounded for new frames. The power consumed by the crosscorrelator becomes in this case the main drawback. In addition, the signal in Figure 6.17b is very much affected by fading channel conditions because of the shortness of the reference signal used in the crosscorrelator, even if the full crosscorrelator is used. In this case, the appearance of secondary peaks helps to hidden the main peaks and the time positioning is no longer a trivial issue.

The block diagram showed in Figure 6.18 represents an alternative solution for the synchronizer as given in [Schwoe01]. In this solution the normalized carrier frequency offset is supposed to be in the range ±0.75 and the estimate ε is obtained as an average of α_1 and α_2, which provide estimations in the range ±2.0 and ±1.0, respectively. The interesting point in this architecture is the massive hardware allocated for the frame detection procedure. Two kinds of signals are defined for this purpose, $C_i(k)$ and $P_i(k)$ given by

$$C_i(k)=\sum_{l=0}^{\mathrm{N}_{\mathrm{avg}}^i-1} r^*(l-k)\cdot r(l-k-\mathrm{N}_{\mathrm{d}}^i)\,, \tag{6.13a}$$

$$P_i(k)=\sum_{l=0}^{\mathrm{N}_{\mathrm{avg}}^i-1} r^*(l-k)\cdot r(l-k)\,, \tag{6.13b}$$

where the different assignments for $\mathrm{N_d}^i$ and $\mathrm{N_{avg}}^i$ are as indicated in Table 6.2.

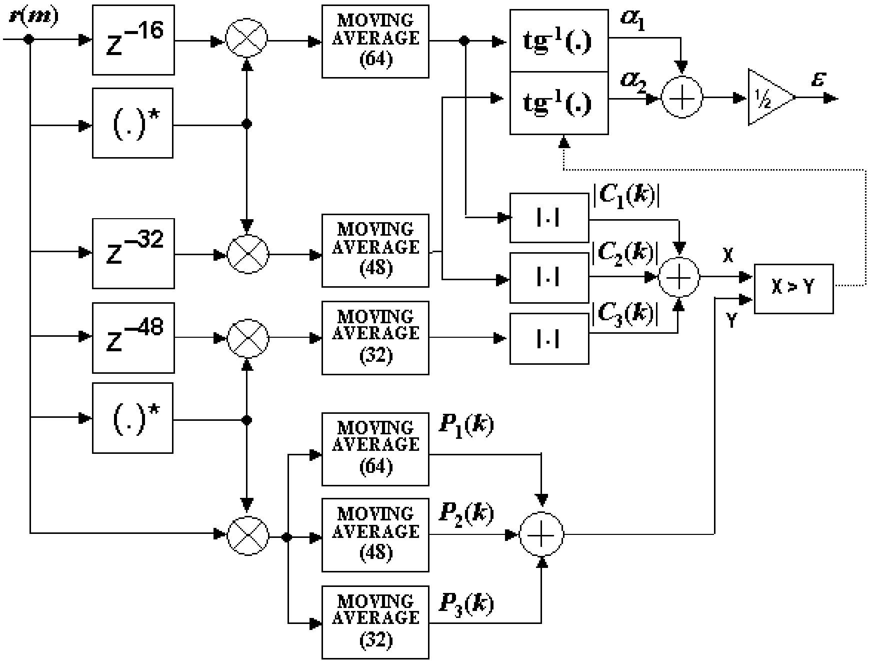


Figure 6.18. Alternative #2 for the synchronizer according to [Schwoe01].

Table 6.2. Assignments for $\mathrm{N_d}^i$ and $\mathrm{N_{avg}}^i$ versus i.

i	$\mathrm{N_d}^i$	$\mathrm{N_{avg}}^i$
1	16	64
2	32	48
3	48	32

The signals $|C_1(k)|$, $|C_2(k)|$ and $|C_3(k)|$ together with $P_1(k)$, $P_2(k)$ and $P_3(k)$ are represented in Figure 6.19. As it can be seen, all these signals show a plateau due to the periodic structure of the preamble symbols. Based on this, the frame detection is asserted when the following condition is satisfied [Schwoe01],

$$|C_1(k)| + |C_2(k)| + |C_3(k)| \geq P_1(k) + P_2(k) + P_2(k)\,, \tag{6.14}$$

and the carrier frequency estimation is computed at the instant k where the condition in (6.14) becomes true. Although robust, the solution in [Schwoe01] requires a big number of power intensive circuits for frame detection, which is a counter-productive feature in wireless applications.

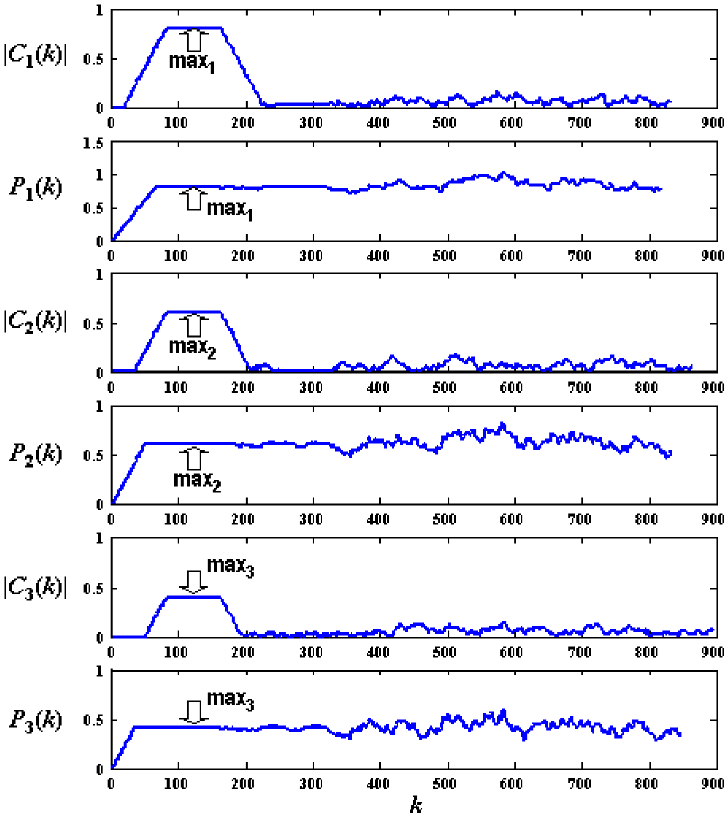


Figure 6.19. Signals used for frame detection in the algorithm proposed in [Schwoe01].

6.5 Performance Evaluation of the Proposed Synchronizer

Since the attenuation suffered by the transmitted OFDM frame is unknown at the receiver, a block able to apply a variable amplification is mandatory prior to digital conversion. This block should be capable to keep the signal inside a certain voltage range given by the bias voltage of the ADCs. The block performing this operation is called Automatic Gain Control (AGC) and a simplified model is depicted in Figure 6.20. Basically it consists of a Gain Controlled Amplifier (GCA) with an amplification driven by the control signal v_C. The gain follows an exponential function with respect to v_C, i.e. the more negative is the control voltage v_C, the bigger is the gain applied by the amplifier. Obviously, this is only true up to a certain value of amplification from which on saturation occurs. The instantaneous power of the signal at the output of the GCA is measured and compared with a reference (V_R) through a differential amplifier. The output of the differential amplifier is therefore low-pass filtered to obtain the control voltage v_C. In the proposed model, the gain control voltage can be sampled and stored in a digital register in order to “freeze” the amplification. By this means, amplification noise can be eliminated.

The frame detector should be robust against two main effects caused by the AGC:

1. Since the AGC is not able to distinguish the signal of interest from the noise, in the absence of any signal the noise will be amplified to the voltage limits of the ADCs as shown in Figure 6.21b. These high noise levels should not provoke false frame detections at the Synchronizer.
2. In a high SNR situation, the AGC has to change very quickly from a high amplification level to a lower one when the signal is received. Since the AGC cannot react instantaneously to sudden changes in the input power level, the AGC output signal will be heavily saturated for a certain time, as shown in Figure 6.22b.

A preliminary test has to determine whether the concept suggested in §6.3.2 for plateau detection still makes sense to be applied when the preamble symbols are affected by an AGC. Figure 6.23 shows a realization of the signals $|J_F(k)|^2$ and $J_{diff}(k)$ for two different values of SNR. As it can be seen, a plateau is no longer visible in $|J_F(k)|^2$ but still the signal $J_{diff}(k)$ is able to provide a prominent peak that is useful for frame detection. Nevertheless, the differentiated signal $J_{diff}(k)$ is very irregular and does not show a clear falling slope as it is the case in Figure 6.7. Therefore, the falling slope detector (or group peak detector) in Figure 6.6 had to be deactivated during simulation and only the instantaneous peak detector was used. Furthermore, in order to reduce the False Alarm Probability (FAP) at low SNR, a 20% threshold was set at the output of $|J_F(k)|^2$.

Simulation results for the FAP are shown in Figure 6.24, in which the channel model A given in Table D.1 is used in all the cases. The model used for the AGC is as in Figure 6.20, where the GCA is supposed to introduce amplitude distortions (saturation) but no phase distortions into the signal. The filter parameters in the feedback loop of the AGC where selected in order to achieve a settling time in which approximately 64 samples (3.2 μs) of the preamble symbols were completely saturated at SNR = 35 dB. Simulations were carried out considering a normalized frequency offset $\varepsilon = 1.2$.

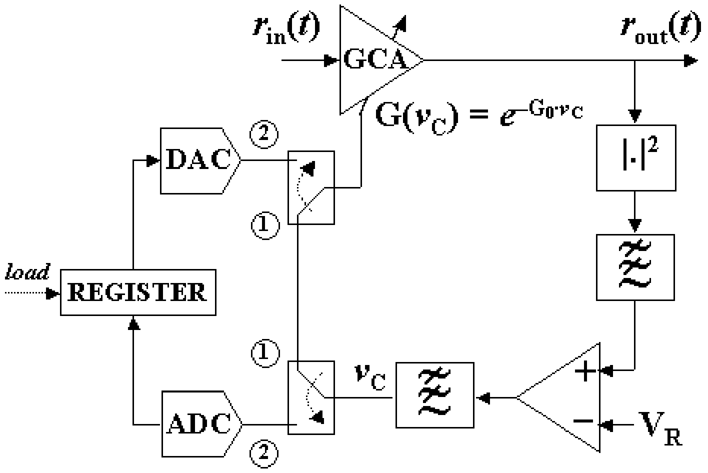
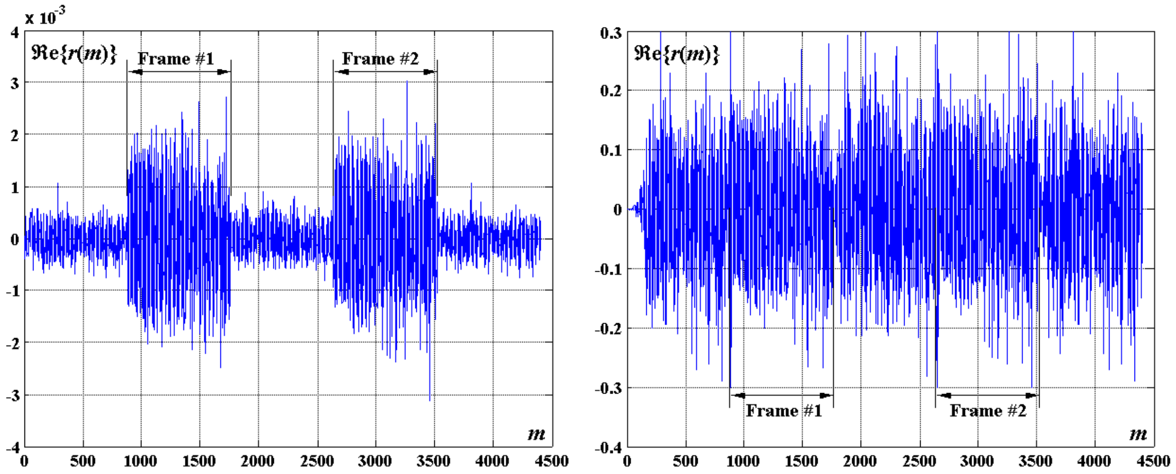
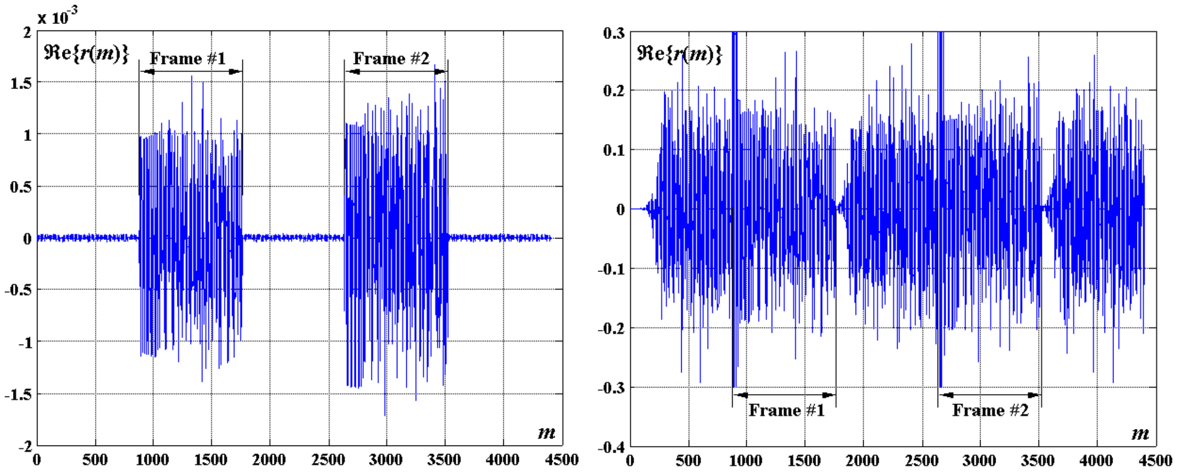


Figure 6.20. Baseband simulation model for the Automatic Gain Control with analog/digital control voltage loop.



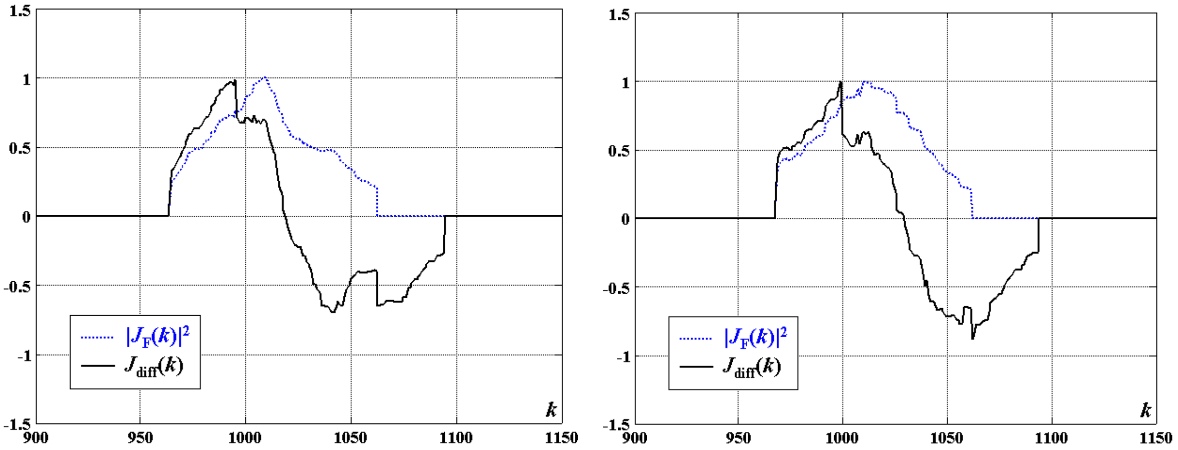
(a) (b)

Figure 6.21. Results for SNR = 10 dB: a) Original attenuated signal (real part only); b) Signal after AGC.



(a) (b)

Figure 6.22. Results for SNR = 35 dB: a) Original attenuated signal (real part only); b) Signal after AGC.



(a) (b)

Figure 6.23. Differentiated signal when preamble is affected by AGC (Threshold = 0.2): a) SNR = 10 dB; b) SNR = 35 dB.

In the definition of FAP used in our simulations, a frame was considered to be correctly detected if the detected starting point was inside a range of ±16 samples from the "ideal" point (i.e. when no AGC and no channel model are used). For the case in which the AGC is used, the FAP is seen to decrease with increasing SNR until a certain value of SNR is reached. From this point on, the distortion due to saturation becomes the dominant effect on the preambles and the FAP degrades as the SNR increases. In order to mitigate the impact of saturation, a small variation in the synchronizer was considered [Kothe03]. Since saturation is easily detected by the ADCs, all those samples being saturated were set to zero before being delivered to the autocorrelator. As a result, a significant reduction of FAP can be obtained for high SNR according to Figure 6.24.

Figure 6.25 depicts the obtained standard deviation for the normalized frequency offset estimator. In all the cases the minimum achievable bound is around 0.01. This value will help in determining the number of bits necessary to represent the frequency offset in the Arctangent calculator (Appendix A).

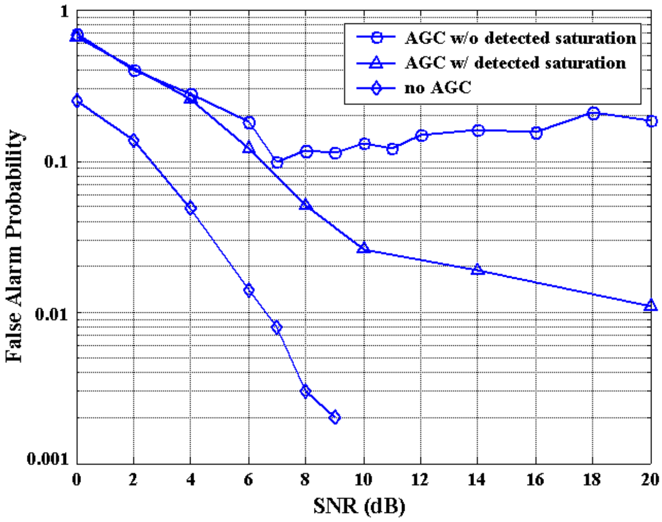


Figure 6.24. Simulated FAP for the proposed frame detector (channel model A, $\varepsilon = 1.2$).

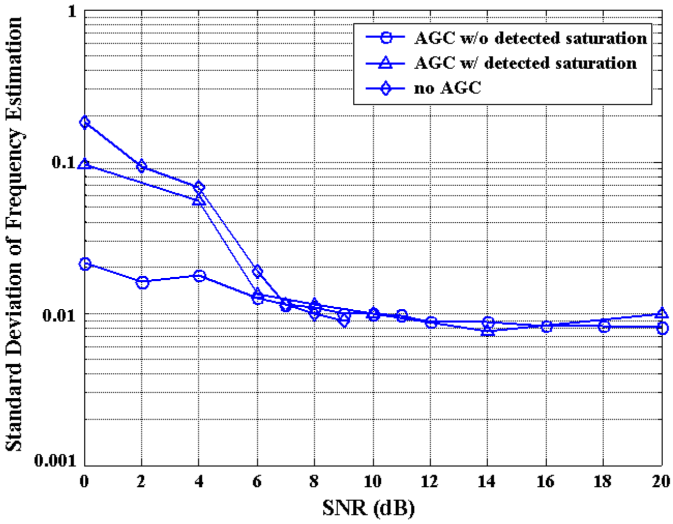


Figure 6.25. Simulated standard deviation for the frequency estimator (channel model A, $\varepsilon = 1.2$).

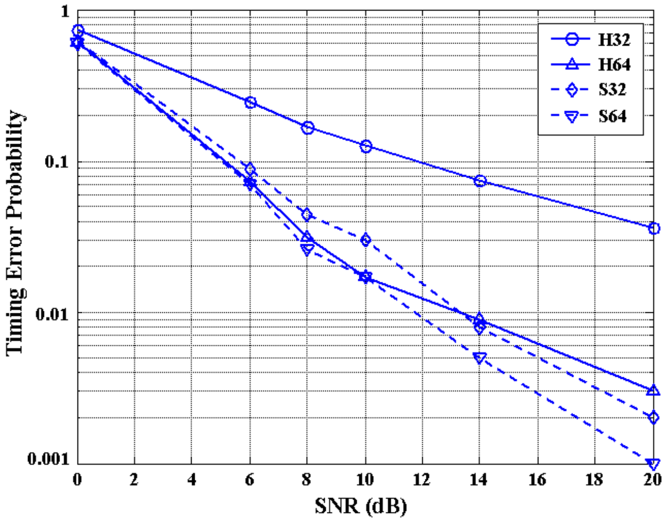


Figure 6.26. Simulated TEP at the output of the crosscorrelator (channel model A, $\varepsilon = 1.2$).

Figure 6.26 depicts the Timing Error Probability (TEP) at the output of the crosscorrelator in Figure 6.14, when a channel model A is used together with a normalized frequency offset $\varepsilon = 1.2$. The symbol timing is determined by the position of the first peak m_1 in Figure 6.15. The ideal value is known beforehand and a timing error is asserted when the estimated position of the peak differs in more than ±2 samples from the ideal position. Nevertheless, this definition for the timing error only makes sense if the cyclic prefix of the symbol being received immediately after the preamble symbols, e.g. SIGNAL symbol in the 802.11a standard, is considered to be 14 samples long instead of 16 in order to compensate positive timing errors.

Four possible versions of the crosscorrelator have been tested depending on the length of the reference signal $c_{REF}(m)$, either 32 or 64 samples, and the type of multiplier, either Hard (XNOR-based) or Soft (floating-point). The peak detection mechanism used in these simulations is not based on a threshold, but considers that the peak is allocated inside a certain time window. Hence, once the frame is detected and the crosscorrelator is activated, the first 32 samples coming out from the crosscorrelator are discarded. Afterwards, the search of the main peak is very similar to the instantaneous peak detection shown in Figure 6.8. However, in this case we apply a 5-bit counter, resulting in a 32-sample observation window. In the simulations it is also considered that the AGC is "frozen" when the crosscorrelator is activated by loading the value of the control signal v_C into a register (see Figure 6.20). The results shown in Figure 6.26 indicate that the selection of a 32-sample Hard crosscorrelator may not be enough and should be increased to 64 samples. Furthermore, there is no significant difference between the Soft and the Hard crosscorrelator as long as the reference is long enough. Despite of these results, our first version of the Synchronizer considers only a reference of 32 samples in order to reduce latency as much as possible.

A last point to consider in this section is the applicability of the proposed solution to the HL2 preamble symbols shown in Figure 6.2. After inspection of these preambles one concludes that our scheme can be applied straightforward to the *uplink burst with long preamble* or the *direct link burst* depicted in Figure 6.2d. With minor modifications, it can also be applied to the *downlink burst* in Figure 6.2b when a carrier frequency offset estimation is already available, e.g. obtained from previous frames, and the AGC is frozen.

Nevertheless, for the other preambles shown in Figures 6.2a/c the concept exposed here is no longer valid. Hence, the *uplink burst with short preamble* in Figure 6.2c does not contain enough short preamble symbols in order to apply our solution. However, the most important preamble is the *broadcast burst* in Figure 6.2a, which is transmitted every 2 ms by the AP, meaning that the MT has to carry out AGC initialization and frequency estimation at intervals of 2 ms. Unfortunately, our solution is not compatible with this preamble structure due to the inverted sections found there, i.e. the sequences IA in Figure 6.2a. With these preambles, the plateaus shown in Figure 6.5a/b no longer overlap in time and therefore, estimations for α and β cannot be performed in parallel.

6.6 Summary

In this Chapter a complete solution for the synchronizer architecture in the IEEE 802.11a standard has been given. The design has been optimized to show low power and low latency characteristics. The low power feature has been achieved by carefully exploiting the preamble structure and by finding smart ways to realize the different operations. Nevertheless, the whole structure has not been specified in this Chapter. More details on some particular components of our Synchronizer are given later on in Chapter 8 and Appendix A.

The low latency feature has been achieved by finding an architecture in which no data buffering is required at all. This comes at the expense of some performance penalties. Hence, the TEP is relatively high due to the short length of the reference signal inside the crosscorrelator. Furthermore, a 3 dB loss in the SNR occurs since the reference CTF is not obtained as an average of the two available long preamble symbols. Nevertheless, this loss is compensated in the next processing stages by applying the frequency direction filter shown in §5.3.2.

Chapter 7

Improved Decision-Directed Channel Estimation for OFDM-WLAN

7.1 Introduction

ONCE THE STRUCTURE OF THE SYNCHRONIZER HAS BEEN DEFINED, it is time to develop solutions for the channel estimation and equalization problems in the special case of OFDM-WLAN systems. A suitable solution was already analyzed in Chapter 5, when dealing with the channel estimation in OFDM systems. The solution proposed here is a modified version of the CD3 channel estimator firstly introduced in [Mig96]. The CD3 is a decision-directed method, whose main advantage lies on the fact that pilot sub-carriers are no longer necessary, thus increasing the amount of information transmitted on each OFDM symbol.

However, there are a number of issues not considered in [Mig96], that make pilot sub-carriers truly necessary. After FFT calculation, a *residual phase error* generally remains in the modulated data, which is due to several factors:

1) Phase variations due to an erroneous frame/symbol timing estimation.
2) Phase variations due to erroneous carrier frequency estimation.
3) Phase variations due to the phase noise.
4) Phase variations due to an offset in the sampling clock frequency during Analog-to-Digital Conversion.

In Chapter 3 it is shown that all these impairments are visible inside an OFDM symbol as a linear phase. Pilot sub-carriers inside an OFDM symbol may help to easily estimate and compensate the remaining linear phase if they were used during the channel estimation, because this phase would be seen as a part of the channel itself. In the standards under consideration the pilots are not intended for channel estimation, i.e. the remaining linear phases should be explicitly estimated using the pilots. A straightforward solution requires an Arctangent calculation to calculate the linear phase at the pilot tones, together with an NCO to correct for it.

Both operations may be realized using computationally intensive algorithms (see Appendix A). In the solution proposed in this Chapter, the linear phase is reduced to a nearly constant phase throughout the OFDM symbol and thus, it allows a significant simplification of the whole operation. Some of the developments presented in the next lines have been published by the author in [Troya01] and [Troya03].

The Chapter starts in §7.2 with a description of the various contributors to the Residual Phase Error (RPE) as well as with a simple signal model, in which all these contributions are included. In §7.3, the CD3 channel estimator in Chapter 5 is re-examined and is found to provide a simple method to estimate and correct the RPE. Nevertheless, this method is “a posteriori” in the sense that it solely deals with the RPE, without trying to directly compensate those impairments, which are on the origin of the RPE itself. In Chapter 3 it is seen that all the contributors listed in §7.2 not only generate an RPE but also, since orthogonality is no longer preserved, ICI and ISI noise components will be added to the signal of interest. Specially critical is the added ICI due to symbol timing errors. Hence, §7.4 deals with the description of a Digital Timing Loop (DTL) which is able to compensate timing errors derived not only from incorrect timing initializations but also from the variable timing originated by sampling clock frequency errors. The DTL represents a solution for the DPLL introduced in §4.3.4 (see Figure 4.13). The performance of the proposed methods is analyzed through computer simulations in §7.5. Once the design issues have been solved, §7.6 addresses the implementation of the most critical blocks found in the channel estimator. These implementation issues are of major importance later on in Chapter 8 when dealing with the VLSI realization of a BBP compliant with the IEEE 802.11a standard. The present Chapter ends with a summary of results in §7.7.

7.2 Contributors to the Residual Phase

7.2.1 Erroneous Frame Timing Estimation

During synchronization the very first sample of the frame has to be estimated for proper data decoding. As we are working on sampled digital data, the resolution in the determination of this sample will strongly depend on the sampling interval, i.e. the inverse of the sampling frequency. Naturally, we are restricted to estimate timing errors, which are a multiple of this sampling interval. The uncertainty in the estimation will be therefore $\pm 0.5T_S$, where T_S is the sampling period. As explained in §3.3, a timing error appears as a linear phase error after performing the FFT. This phase error will be the same for all the OFDM symbols in that frame. In addition to this, an ICI component will appear due to the loss of orthogonality, thus increasing the noise content inside the OFDM symbol. In the standards under consideration a timing error in the order of half the sampling interval already results in a substantial degradation of the ISR (see Figure 3.4).

7.2.2 Erroneous Carrier Frequency Estimation

In a typical practical scenario, during RF down-conversion, the different oscillators are not exactly tuned to the expected frequencies. The *preamble symbols* are used to estimate the carrier frequency offset, as explained in Chapter 6. Nevertheless, a perfect frequency estimation cannot be expected due to noise, multipath propagation, AGC effects,… Simulation results obtained in Chapter 6 show that the estimates of the frequency offset are affected by an error of ±1% on average. This small residual carrier frequency error generates a constant phase inside the OFDM symbol after FFT (see §3.2). However, unlike the case of the timing error, this phase gets accumulated from symbol to symbol, thus becoming a large phase after several symbols. Furthermore, ICI will occur due to the loss of orthogonality.

7.2.3 Phase Noise

The several PLLs used during RF down-conversion will generate phase noise. There are mainly two effects associated with the phase noise: Inter-Carrier Interference (ICI) and Common Phase Error (CPE). Interestingly, both ICI and CPE depend on the number of sub-carriers N used by the OFDM system but in an inverse way. The bigger N, the greater is the ICI power, but the smaller the CPE, and vice versa (§3.5).

The ICI appears as an additive Gaussian noise and the only way to combat it is by improving the RF oscillators. The CPE is seen as a constant phase inside the OFDM symbol, similar to the effect mentioned above when a residual carrier frequency offset is present. Nevertheless, in this case the constant phase is not getting accumulated, but changes randomly from one symbol to the next one. In the standards under consideration, 64 sub-carriers are used, and the CPE effect will therefore dominate above the ICI. This is different for other OFDM systems like DAB [DAB] or DVB [DVB], where thousands of sub-carriers are used.

7.2.4 Sampling Clock Frequency Error

In a real implementation, the system is designed to sample the analog input signal at a certain frequency (f_S). However, the oscillator will introduce some error in f_S. In the standards considered here, 80 samples per symbol are expected, before the FFT and cyclic prefix extraction, with $f_S = 20$ MHz. In the case of a sampling oscillator with e.g. 20 ppm frequency error, this turns into $f_S = 20000400$ Hz. Thus, 80.00002 samples are obtained for the initial symbol instead of exactly 80, i.e. a timing error of 0.00002 samples. This timing error is not fixed, but it will be 0.00004 samples for the next symbol, 0.00006 for the third one and so on. In essence, the sampling clock frequency error will be seen as a *dynamic* timing error. Previously it has been explained that a timing error generates a linear phase error after FFT. In this case, the slope of this linear phase error will change from symbol to symbol (see §3.4).

7.2.5 Combination of Errors

As a result, sub-carrier k belonging to symbol l can be expressed as follows after FFT calculation

$$Y_{k,l} = A_{k,l} \cdot H_{k,l} \cdot e^{j \cdot \phi_l(k)} + V_{k,l} \,, \tag{7.1}$$

where $A_{k,l}$ are the modulated data (M-QAM scheme), $H_{k,l}$ represents the CTF coefficient affecting sub-carrier k of symbol l and $V_{k,l}$ are the samples of a zero-mean Gaussian noise process, which also includes the ICI. The phase component $\phi_l(k)$ distorting the modulated data $A_{k,l}$ is as follows, considering what it was stated above:

$$\phi_l(k) = (m_l \cdot k) + c_l \,, \tag{7.2}$$

where the slope m_l in symbol l can be expressed as

$$m_l = m_0 + (l \cdot \xi). \tag{7.3}$$

The parameter ξ relates to the sampling clock error and may be positive or negative, depending on whether the sampling is faster or slower than expected. The value m_0 will be given by an error in the symbol timing estimation and is constant throughout the symbols. The term c_l in (7.2) can be further decomposed for symbol l as

$$c_l = (l \cdot c_0) + \alpha_l , \tag{7.4}$$

where α_l is the contribution of the CPE for that particular symbol (random value) and c_0 is the phase derived from the residual carrier frequency offset, which is accumulated from symbol to symbol. The foregoing model will be used in the next section to derive a simplified RPE estimation and correction mechanism.

7.3 Residual Phase Error: Estimation and Correction

7.3.1 The CD3 Channel Estimator Revisited

Since pilots cannot be used for channel estimation in the particular case of the 802.11a and HL2 standards, a decision-directed channel estimator as given in §5.3 was selected for implementation. Mignone and Morello firstly proposed the CD3 scheme in [Mig96]. However, unlike to their solution, the use of a small number of pilots will help in the estimation of the RPE originated by the several contributors mentioned in §7.2.

A block diagram of the modified CD3 is shown in Figure 7.1. The estimator is designed in such a way that the samples of symbol l are used to calculate an estimation of the channel, which will be used to correct the symbol l+L, where L is the delay introduced by the feedback loop (in symbols). The value L depends very much on the traceback length of the Viterbi decoder in the FEC block (see analysis presented in §5.3.5).

In the scheme shown in Figure 7.1, the data are delivered at the output of the FFT following the order given in §6.3.5, in which the pilots are positioned first followed by the data, from sub-carrier –26 to +26, except for the DC and NULL sub-channels. Furthermore, the pilots are already sign-corrected according to the standards (see §2.4.1). In Figure 7.1, data and pilot sub-carriers are represented separately for better understanding. The channel estimator is operated in the following steps:

1. The very first symbol obtained from the synchronizer will be the reference symbol. This symbol is derived from the *long preamble symbols* as explained in §6.3.5. The reference $H_{k,\mathrm{REF}}$ undergoes the frequency direction filter as given in §5.3.2 and is finally stored in an estimation buffer. Note that the time direction filter shown in Figure 5.11 is no longer considered in this solution, i.e. the parameter κ in (5.58) is permanently set to zero.
2. The first symbol coming after the reference symbol, the so-called SIGNAL symbol, is corrected by the equalization block performing a division between this symbol and the reference symbol stored in the estimation buffer.
3. The channel-corrected symbol will be phase corrected, fully demodulated and decoded.
4. The bits obtained from the SIGNAL symbol at the output of the Viterbi decoder will give information about the *rate* and the *length* of the frame (in number of bytes). This information is used to initialize the feedback loop but not inserted back into the loop.
5. The first DATA symbol (symbol #1) coming after the SIGNAL symbol will be corrected using again the reference channel estimation.

6. The resulting symbol will be phase corrected, demodulated and decoded.
7. The output bits corresponding to the symbol #1 will be fed back into the channel estimation block.
8. The *mapper* block in the feedback loop provides 48 complex samples per symbol, $\hat{A}_{k,l-\mathrm{L}}$, corresponding to sub-carriers –26,…,+26, except for the pilots, DC and NULL sub-channels. These samples are divided by the input complex samples $Y_{k,l-\mathrm{L}}$ of the same symbol prior to equalization, thus obtaining a new channel estimation $\hat{H}_{k,l-\mathrm{L}}$. By this procedure, any residual phase contained in $Y_{k,l-\mathrm{L}}$ will remain in the channel estimation, $\hat{H}_{k,l-\mathrm{L}}$, since the samples $\hat{A}_{k,l-\mathrm{L}}$ are free of any phase error. The vector $\hat{H}_{k,l-\mathrm{L}}$ is merged together with the sign-corrected pilots, $P_{k,l-\mathrm{L}}$, at the input of the frequency direction filter, thus generating a vector of 52 complex samples containing sub-carriers -26,…,+26 (without DC). This filter is used for noise reduction and is based on the low-rank approximation (see §5.3.2 for further details).
9. The obtained channel estimation in the frequency domain is again separated into two vectors, $\tilde{H}_{k,l-\mathrm{L}}$ and $\tilde{P}_{k,l-\mathrm{L}}$, containing channel estimations for the data and pilot sub-carriers, respectively. The former is used directly for equalization of the data sub-carriers. The later is employed to generate the vector $P_{k,l}^{\phi}$ applied to the *residual phase correction* block. In this case, the channel is supposed not to change significantly during a period of L symbols, so that after equalization, the vector $P_{k,l}^{\phi}$ will be approximately a phasor with normalized magnitude and a phase given by (7.2).
10. For the next incoming symbol, the operation will be the same starting from point 5.

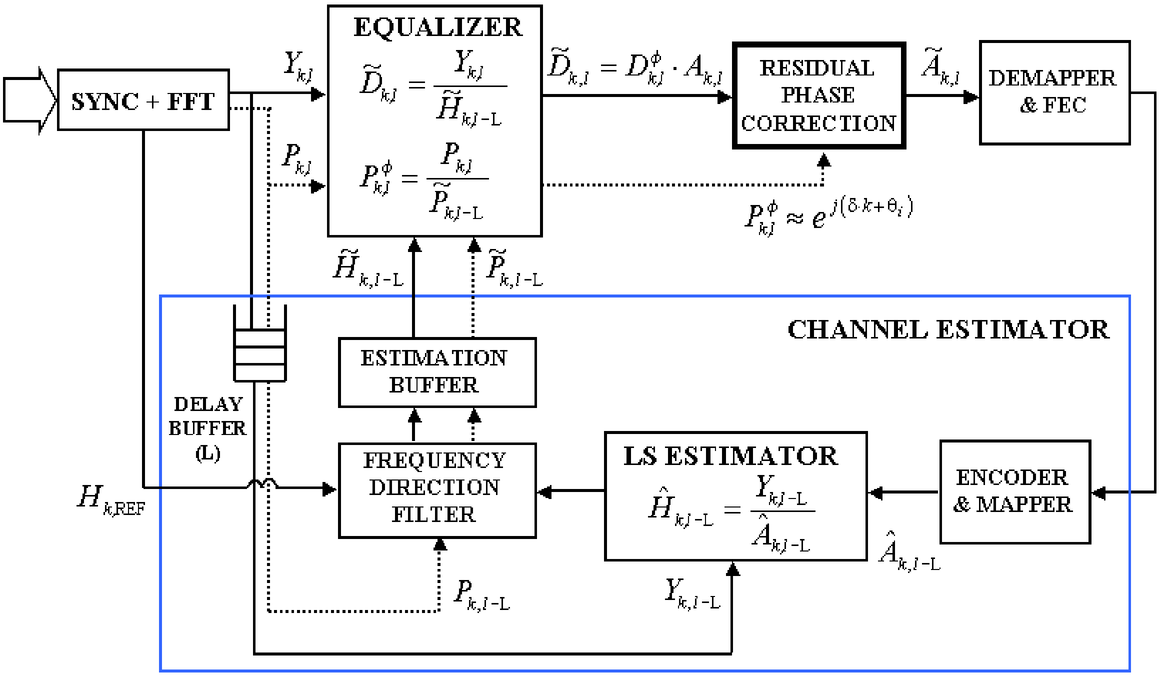


Figure 7.1. Block diagram of the modified CD3 channel estimator including the RPE estimator.

7.3.2 Phase Estimation

According to the scheme in Figure 7.1, the channel estimation used to correct symbol l will be affected by the phase error of symbol l–L. After division (equalization), the symbol l will retain an RPE of the form

$$D^{\phi}_{k,l} = e^{j\cdot(\phi_l(k)-\phi_{l-\mathrm{L}}(k))}. \tag{7.5}$$

The phase difference in (7.5) is as follows considering (7.2), (7.3) and (7.4)

$$\phi_l(k) - \phi_{l-\mathrm{L}}(k) = \mathrm{L}\cdot\xi\cdot k + \mathrm{L}\cdot c_0 + (\alpha_l - \alpha_{l-\mathrm{L}}), \tag{7.6}$$

which results in a linear residual phase whose slope does not depend on l. In the same way, the pilots in symbol l are divided by the pilots in symbol l–L. In any case, the channel is supposed not to change significantly during a period of L symbols, so that after division, the resulting pilots $P^{\phi}_{k,l}$ will be phasors with normalized magnitude and a phase given by (7.6). In the absence of noise, this phasor may be expressed as

$$P^{\phi}_{k,l} \approx e^{j(\delta\cdot k+\theta_l)}, \tag{7.7}$$

where $\delta = \mathrm{L}\cdot\xi$, $\theta_l = \mathrm{L}\cdot c_0 + (\alpha_l - \alpha_{l-\mathrm{L}})$ and $k = -21, -7, +7, +21$.

The method proposed here for the phase estimation considers that the condition $|\delta\cdot k| << 1$ is satisfied $\forall k\in[-26, +26]$. In this case, the expression in (7.7) may be simplified by considering a first order approximation of the complex exponential, yielding

$$P^{\phi}_{k,l} = \cos(\theta_l) - \delta\cdot k\cdot\sin(\theta_l) + j\big(\sin(\theta_l) + \delta\cdot k\cdot\cos(\theta_l)\big). \tag{7.8}$$

In the previous expression four parameters are of interest, namely $\cos(\theta_l)$, $\sin(\theta_l)$, $\delta\cdot\cos(\theta_l)$ and $\delta\cdot\sin(\theta_l)$. These parameters can be obtained combining the four available pilots. The former two may be expressed as

$$\cos(\theta_l) = (1/4)\cdot(\Re\mathrm{e}\{P^{\phi}_{-21,l}\} + \Re\mathrm{e}\{P^{\phi}_{-7,l}\} + \Re\mathrm{e}\{P^{\phi}_{+7,l}\} + \Re\mathrm{e}\{P^{\phi}_{+21,l}\}), \tag{7.9}$$

$$\sin(\theta_l) = (1/4)\cdot(\Im\mathrm{m}\{P^{\phi}_{-21,l}\} + \Im\mathrm{m}\{P^{\phi}_{-7,l}\} + \Im\mathrm{m}\{P^{\phi}_{+7,l}\} + \Im\mathrm{m}\{P^{\phi}_{+21,l}\}), \tag{7.10}$$

where $\Re\mathrm{e}\{.\}$ denotes the real part and $\Im\mathrm{m}\{.\}$ the imaginary part. Note that (7.9) and (7.10) require no multiplication, but just addition and one shift operation.

In the determination of $\delta\cdot\cos(\theta_l)$ and $\delta\cdot\sin(\theta_l)$ another approximation is considered, again trying to avoid any multiplication. From (7.8) it is obtained that

$$35\cdot\delta\cdot\sin(\theta_l) = (1/2)\cdot(\Re\mathrm{e}\{P^{\phi}_{-21,l}\} + 2\cdot\Re\mathrm{e}\{P^{\phi}_{-7,l}\} - 2\cdot\Re\mathrm{e}\{P^{\phi}_{+7,l}\} - \Re\mathrm{e}\{P^{\phi}_{+21,l}\}), \tag{7.11a}$$

$$28\cdot\delta\cdot\sin(\theta_l) = (1/2)\cdot(\Re\mathrm{e}\{P^{\phi}_{-21,l}\} + \Re\mathrm{e}\{P^{\phi}_{-7,l}\} - \Re\mathrm{e}\{P^{\phi}_{+7,l}\} - \Re\mathrm{e}\{P^{\phi}_{+21,l}\}). \tag{7.11b}$$

Considering the elementary decompositions 36/32 = 1 + 1/8 and 28/32 = 1 – 1/8, and approximating the factor 35 in (7.11a) by 36, one may obtain the following simplification,

$$\delta\cdot\sin(\theta_l)\cdot\left(\frac{35}{32}+\frac{28}{32}\right) \approx \delta\cdot\sin(\theta_l)\cdot\left(\frac{36}{32}+\frac{28}{32}\right) = 2\cdot\delta\cdot\sin(\theta_l)\,. \tag{7.12}$$

Inserting (7.11a) and (7.11b) into (7.12), yields

$$\delta\cdot\sin(\theta_l) \approx (1/128)\cdot(2\cdot\Re\mathrm{e}\{P^{\phi}_{-21,l}\} + 3\cdot\Re\mathrm{e}\{P^{\phi}_{-7,l}\} - 3\cdot\Re\mathrm{e}\{P^{\phi}_{+7,l}\} - 2\cdot\Re\mathrm{e}\{P^{\phi}_{+21,l}\}). \tag{7.13}$$

The exact value of the previous expression is $(126/128)\cdot\delta\cdot\sin(\theta_l)$, thus the simplification done in (7.12) only introduces an error of 1.6% into (7.13). The whole calculation can be still realized using shift operations and additions. Equivalently, similar simplifications can be done in order to obtain $\delta\cdot\cos(\theta_l)$ as

$$\delta \cdot \cos(\theta_l) \approx -(1/128) \cdot (2 \cdot \Im\mathrm{m}\{P^{\phi}_{-21,l}\} + 3 \cdot \Im\mathrm{m}\{P^{\phi}_{-7,l}\} - 3 \cdot \Im\mathrm{m}\{P^{\phi}_{+7,l}\} - 2 \cdot \Im\mathrm{m}\{P^{\phi}_{+21,l}\}) . \quad (7.14)$$

Note that in case any of the pilots being severely distorted by big channel attenuation, this effect can be partly compensated since the parameters of interest are obtained as averaged values of the real and imaginary parts of $P^{\phi}_{k,l}$.

7.3.3 Phase Correction

Once the previous four parameters have been determined, the correction of the data sub-channels is straightforward. The first data after the pilots is $\tilde{D}_{-26,l}$. The phase contribution found in this particular sub-carrier due to the residual phase error $D^{\phi}_{-26,l}$ can be written as

$$D^{\phi}_{-26,l} = \cos(\theta_l) + 26 \cdot \delta \cdot \sin(\theta_l) + j\big(\sin(\theta_l) - 26 \cdot \delta \cdot \cos(\theta_l)\big) , \quad (7.15)$$

and therefore the corresponding sample in the data path should be multiplied by the complex conjugate of (7.15) in order to compensate its RPE.

The factors $\delta \cdot \sin(\theta_l)$ and $-\delta \cdot \cos(\theta_l)$ are multiplied by 26 in (7.15), but actually no multiplication is really needed since $26 = 2^5 - 2^2 - 2$. Furthermore, this is only necessary for the first sample; for the sample at $k = -25$, the corresponding correcting factor $(D^{\phi}_{-25,l})^*$ will be obtained as

$$\Re\mathrm{e}\{(D^{\phi}_{-25,l})^*\} = \Re\mathrm{e}\{(D^{\phi}_{-26,l})^*\} - \delta \cdot \sin(\theta_l) , \quad (7.16a)$$

$$\Im\mathrm{m}\{(D^{\phi}_{-25,l})^*\} = \Im\mathrm{m}\{(D^{\phi}_{-26,l})^*\} - (-\delta \cdot \cos(\theta_l)) . \quad (7.16b)$$

Special care has to be taken when calculating $(D^{\phi}_{-20,l})^*$, $(D^{\phi}_{-6,l})^*$, $(D^{\phi}_{+1,l})^*$, $(D^{\phi}_{+8,l})^*$, $(D^{\phi}_{+22,l})^*$. For these particular cases, there is a “hole” left by either the pilots (k = –21, –7, +7, +21) or the DC sub-channel (k = 0), thus yielding

$$\Re\mathrm{e}\{(D^{\phi}_{k,l})^*\} = \Re\mathrm{e}\{(D^{\phi}_{k-2,l})^*\} - 2 \cdot \delta \cdot \sin(\theta_l) , \quad (7.17a)$$

$$\Im\mathrm{m}\{(D^{\phi}_{k,l})^*\} = \Im\mathrm{m}\{(D^{\phi}_{k-2,l})^*\} - (-2 \cdot \delta \cdot \cos(\theta_l)) , \quad (7.17b)$$

with k = –20, –6, +1, +8, +22.

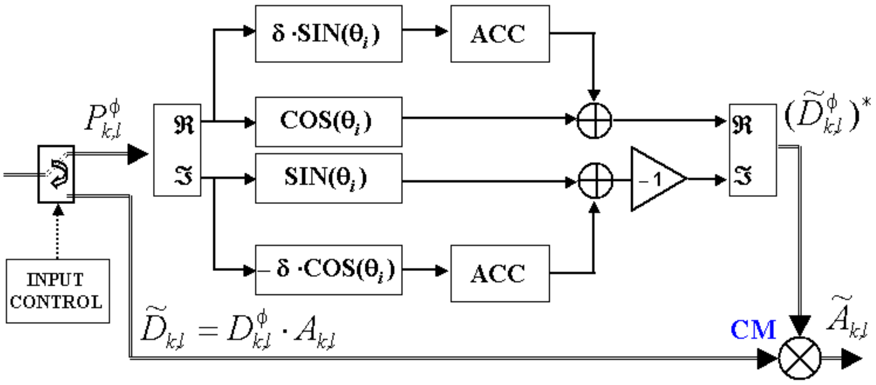


Figure 7.2. Block diagram of the proposed RPE correction mechanism.

In Figure 7.2, a simplified block diagram for implementation of the proposed RPE correction mechanism is presented. The blocks named ACC carry out the operations in (7.16a/b) and (7.17a/b) and are shown in more detail in Figure 7.3. Initially, the register R1 in Figure 7.3 contains the estimated value for $\delta\cdot\sin(\theta_l)$ (or $-\delta\cdot\cos(\theta_l)$) and the switches SW1 and SW2 are both at position "0". The value of R1 multiplied by 26 will be stored in the register R2. After that, SW1 and SW2 will both switch to position "1". The switch SW2 will only change to "2" when $(D^{\phi}_{-20,l})^*$, $(D^{\phi}_{-6,l})^*$, $(D^{\phi}_{+1,l})^*$, $(D^{\phi}_{+8,l})^*$ and $(D^{\phi}_{+22,l})^*$ are to be calculated. In all the cases, the final correction is achieved by performing a single complex multiplication (block CM in Figure 7.2).

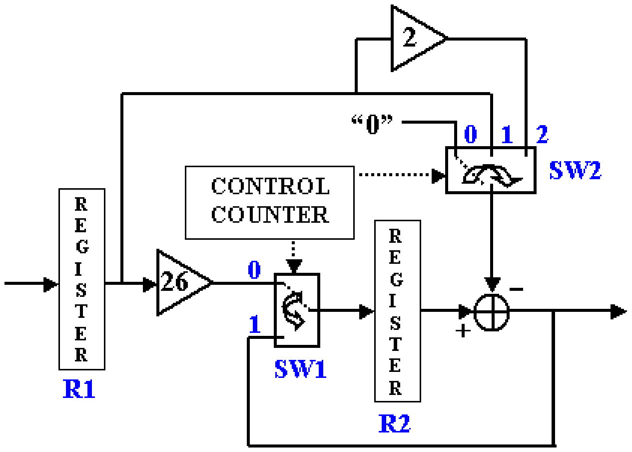


Figure 7.3. Block diagram of the *Accumulators* (ACC) shown in Figure 7.2.

7.4 Timing Adjustment by Variable Interpolation

7.4.1 Motivation

The method shown in §7.3 for phase estimation and correction is *a posteriori*, i.e. no attempt is done to directly compensate the contributors listed in §§7.2.1 to 7.2.4. Instead, the most evident effect, namely the phase error present after FFT, is estimated and corrected. From the analysis presented in Chapter 3, an important amount of noise is also added to the symbols after FFT due to ISI and ICI. In order to reduce the noise, correcting methods prior to FFT calculation should be considered.

From all the contributors described in §7.2, the ones showing the most critical effects are those related to timing errors. The residual carrier frequency error cannot be fully eliminated since it is originated by the digital noise, i.e. the limited number of bits used in the implementation. Regarding to phase noise, this is contributing little added noise. As opposed to this, a timing error may lead to very bad performance if samples of adjacent symbols are processed within the symbol of interest. Already a timing error of half a sample has catastrophic effects in terms of added noise (see Figure 3.4). The sampling clock error also belongs to this group, since its effect is a slow but constant degradation of the timing. After some hundreds of symbols, the accumulated timing error is large enough to cause a break down of the whole receiver.

A timing error discriminator was already presented in §4.3.4 based on the pilot sub-carriers. The error signal generated by this discriminator is used in a tracking system for timing correction. Note that in this solution it is irrelevant where the timing error is generated. The pilots $P^{\phi}_{k,l}$ as given in (7.7) may be used to generate a timing error estimation according to (4.57). How to apply the error signal into a correction mechanism is the main point in this section. Hence, the design of the timing error compensation block and the control block in Figure 4.13 will be treated in the next two sub-sections, respectively.

7.4.2 Principles of Digital Variable Interpolation

The first references discussing the use of digital interpolation applied to timing correction are given in [Gar93] and [Erup93], and further refined in [Meyr97]. The main idea is that a timing correction can be achieved by dynamically increasing or decreasing the sampling rate of a digital sequence. Instead of varying the sampling clock itself, the solution proposed in [Gar93] and [Erup93] considers a method based on purely digital interpolation and decimation. The solution is straightforward when the sampling and symbol rates are related by some rational value. On the contrary, when they are *incommensurate*, a more complex procedure is required.

The error detector produces an error signal at rate $1/\mathrm{T_{SYM}}$ based on $P_{k,l}^{\phi}$, with $\mathrm{T_{SYM}} = \mathrm{T}+\mathrm{T_G}$, using fractionally spaced samples $k\cdot\mathrm{T_I} = k\cdot(\mathrm{T_{SYM}}/\mathrm{M_I})$ (k being the time variable). The parameter $\mathrm{M_I}$ represents the number of samples per symbol used by the error detector. Since the sampling rate $1/\mathrm{T_S}$ is not an exact multiple of the symbol rate, the samples at instants $k\cdot\mathrm{T_I}$ have to be mapped onto the time scale $\{k\cdot\mathrm{T_S}\}$ of the receiver. Mathematically, this is done by expressing the time instant $k\cdot\mathrm{T_I} + \tau_\mathrm{I}\cdot\mathrm{T_I}$ by multiples of $\mathrm{T_S}$ plus fractional rest, i.e.

$$k\cdot\mathrm{T_I} + \tau_\mathrm{I}\cdot\mathrm{T_I} = \lfloor k\cdot\mathrm{T_I} + \tau_\mathrm{I}\cdot\mathrm{T_I}\rfloor\cdot\mathrm{T_S} + \mu_k\cdot\mathrm{T_S} = m_k\cdot\mathrm{T_S} + \mu_k\cdot\mathrm{T_S}, \tag{7.18}$$

where $0 \le \mu_k < 1$ is the *fractional delay* and m_k is the *basepoint*. $\mathrm{T_I}$ represents the ideal sampling time, whereas $\mathrm{T_S}$ is the actual sampling time. The function $\lfloor x\rfloor$ represents the largest integer smaller than or equal to the real value x.

As it is shown in [Gar93], the output sequence $y(k)$ of the timing error compensation block, sampled at intervals $\mathrm{T_I}$, can be obtained after filtering the input sequence $x(m)$, sampled at intervals $\mathrm{T_S}$, through the following expression,

$$\begin{aligned} y(k\cdot\mathrm{T_I} + \tau_\mathrm{I}\cdot\mathrm{T_I}) &= \sum_m x(m\cdot\mathrm{T_S})\cdot h_\mathrm{I}(k\cdot\mathrm{T_I} + \tau_\mathrm{I}\cdot\mathrm{T_I} - m\cdot\mathrm{T_S}) \\ &= \sum_m x(m\cdot\mathrm{T_S})\cdot h_\mathrm{I}((m_k - m)\cdot\mathrm{T_S} + \mu_k\cdot\mathrm{T_S}) \\ &= \sum_i x((m_k - i)\cdot\mathrm{T_S})\cdot h_\mathrm{I}((i+\mu_k)\cdot\mathrm{T_S})\,. \end{aligned} \tag{7.19}$$

The samples of $y(k)$ are referred to as the *interpolants*. Note that an FIR structure has been selected for the interpolation filter in (7.19) and that the filter coefficients $h_\mathrm{I}(i + \mu_k)$ are time-variant. During operation, a new sample is read into the filter at rate $1/\mathrm{T_S}$, while a new output is computed only at the basepoints $m_k\cdot\mathrm{T_S}$ (equivalent to decimation). The value of μ_k is updated each time m_k changes.

The design of optimum variable interpolators has been treated by several authors in the literature [Erup93], [Meyr97], [Laak96], [Kim97]. The filter impulse response can be optimized by minimizing the MSE contingent on selected filter length, SNR and expected possible values of μ_k [Kim97]. However, in high-speed WLAN systems, where the sampling time T_S is generally very small, simple interpolation structures are a must. In this direction, two possible strategies may be followed:

1) Precompute and store impulse response coefficients $h_I(i + \mu_k)$ for all the possible expected values for μ_k.
2) Compute interpolants directly on-line without storing impulse response samples, or possibly even without calculating impulse responses explicitly. As explained in [Erup93], this feature appears to be feasible only for polynomial-based filters. Such an approach will lead to the so-called Farrow structures [Far88].

In a polynomial-based filter it is supposed that each coefficient $h_I(i + \mu) \equiv h_i(\mu)$ can be expressed as

$$h_i(\mu) = \sum_{l=0}^{M(i)} c_l(i) \cdot \mu^l \ , \tag{7.20}$$

and substituting (7.20) into (7.19) yields

$$\begin{aligned} y(k) &= \sum_{i=I_1}^{I_2} x(m_k - i) \cdot \sum_{l=0}^{M} c_l(i) \cdot \mu^l \\ &= \sum_{l=0}^{M} \mu^l \cdot \sum_{i=I_1}^{I_2} c_l(i) \cdot x(m_k - i) \, , \end{aligned} \tag{7.21}$$

where it has been assumed that all the polynomials have the same degree M. Furthermore, for implementation purposes the filter has been reduced to a finite length $I = I_2 - I_1 + 1$. The result in (7.21) suggests that a polynomial interpolator can thus be realized as a bank of M parallel FIR filters, where the output of the l-th branch is firstly multiplied by μ^l and then summed up. This structure was firstly devised by Farrow in [Far88].

Now a method for determining the coefficients $c_l(i)$ in (7.21) has to be conceived. The authors in [Erup93] and [Meyr97] suggest the use of Lagrange polynomial interpolators, since they fit very well the expression given in (7.20). Only two solutions are going to be considered in our further analysis, i.e. the ones derived from first and second order Lagrange polynomial approximations. In this case, the parameter M is directly the value of the polynomial order (M = 1 for 1st order or lineal; M = 2 for 2nd order or parabolic) and the filter length I was selected to be I = M + 1.

A unique parameter γ determines the second order Lagrange polynomial. In this case the coefficients $c_l(i)$ are given in Table 7.1. The values for the first order Lagrange polynomial are obtained by setting $\gamma = 0$ (see Figure 7.4a).

Table 7.1. Farrow coefficients $c_l(i)$ for Lagrange parabolic interpolation.

i	$l = 0$	$l = 1$	$l = 2$
–2	0	$-\gamma$	γ
–1	0	$\gamma + 1$	$-\gamma$
0	1	$\gamma - 1$	$-\gamma$
1	0	$-\gamma$	γ

The analysis presented by the authors in [Erup93] suggests to use $\gamma = 0.5$ in a second order implementation due to its good performance, thus yielding the scheme shown in Figure 7.4b.

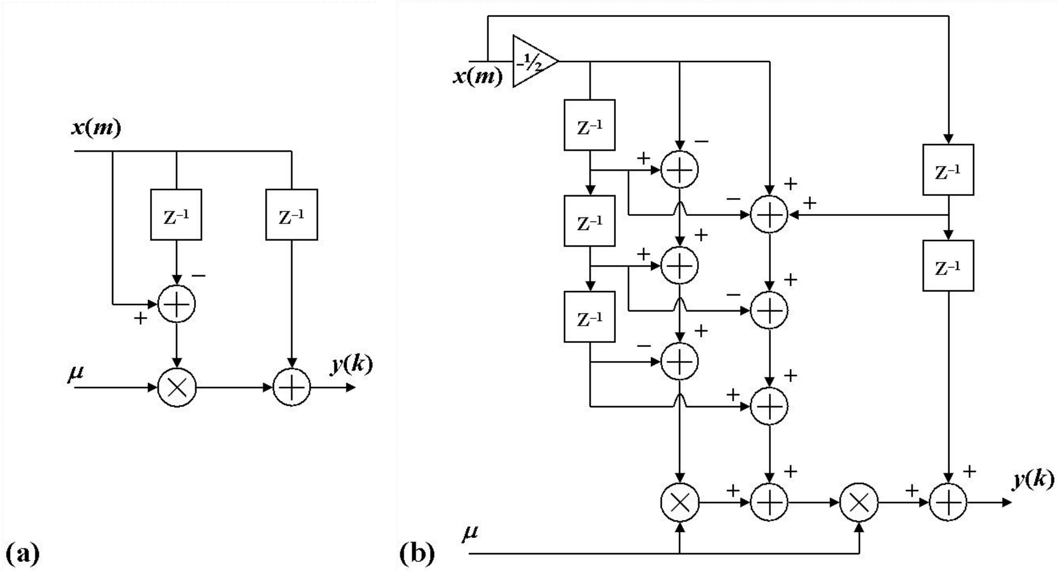


Figure 7.4. Farrow structure for the second order Lagrange interpolator with: a) $\gamma = 0$ (resulting in the linear interpolator); b) $\gamma = 0.5$.

At this point, possible structures for the timing error compensation block implemented as a variable interpolator have been determined. In the next section we will describe how to generate the control parameters m_k and μ_k based on the error signal provided by the timing error discriminator defined in (4.57).

7.4.3 Interpolator Control

The discriminator $S(\Delta t,\delta)$ given in (4.58), where Δt represents the timing error (4.56), was analyzed for the particular case of $\delta = 0.5$. The s-curve depicted in Figure 4.14 shows that the discriminator has a nearly linear response for timing errors in the range ±0.5 samples. Note that this curve is obtained after low-pass filtering the signal $a(l,\Delta t)$ in (4.57). However, the parameter of interest is not the absolute timing error, but the relative error existing between T_I and T_S, i.e. $T_I/T_S = 1 + \chi$. It is straightforward to see that $\chi = \Delta t/(N+N_G)$, where $N+N_G$ is the total number of samples per OFDM symbol.

The control block in Figure 4.13 contains a *control word*, $w(l)$, which is updated on a symbol basis and provides the latest estimate of the ratio T_I/T_S as follows,

$$w(l + 1) = w(l) + K_w \cdot e(l, \Delta t), \tag{7.22}$$

with

$$e(l + 1, \Delta t) = e(l, \Delta t) + K_e \cdot a(l, \Delta t). \tag{7.23}$$

The curve in Figure 4.14 was normalized according to the maximum value in $S(\Delta t,\delta)$, i.e. $S_{\max} = 15.76$ (with $P = 4$, $\Delta = 14$, $\delta = 0.5$). Therefore, the parameter K_w in (7.22) is selected to be

$$K_w = (2 \cdot S_{\max} \cdot (N + N_G))^{-1}. \tag{7.24}$$

The control word will reach in the steady state the value $w(\infty) = 1 + \chi$. The expression in (7.23) represents a first order IIR low-pass filter, whose bandwidth depends on the parameter K_e. This parameter determines how fast the steady state is reached. A value which has proven to give good results is $K_e = 0.01$.

The parameters m_k (basepoint) and μ_k (fractional delay) to be used in the variable interpolator will be recursively computed as explained hereinafter. We already expressed $k \cdot \mathrm{T_I} + \tau_\mathrm{I} \cdot \mathrm{T_I}$ as a function of (m_k, μ_k) in (7.18). The next sample $(k+1) \cdot \mathrm{T_I} + \tau_\mathrm{I} \cdot \mathrm{T_I}$ is given by

$$(k + 1) \cdot \mathrm{T_I} + \tau_\mathrm{I} \cdot \mathrm{T_I} = m_k \cdot \mathrm{T_S} + \mu_k \cdot \mathrm{T_S} + \mathrm{T_I} = m_k \cdot \mathrm{T_S} + (\mu_k + (\mathrm{T_I} / \mathrm{T_S})\ ) \cdot \mathrm{T_S}. \tag{7.25}$$

Replacing in the previous expression the unknown ratio $(\mathrm{T_I} / \mathrm{T_S})$ by its estimate $w(m_k)$ we obtain

$$(k + 1) \cdot \mathrm{T_I} + \tau_\mathrm{I} \cdot \mathrm{T_I} = m_k \cdot \mathrm{T_S} + \lfloor \mu_k + w(m_k) \rfloor \cdot \mathrm{T_S} + [\mu_k + w(m_k)]_{\bmod 1} \cdot \mathrm{T_S}. \tag{7.26}$$

From this, it readily follows the recursion for the estimates,

$$m_{k+1} = m_k + \lfloor \mu_k + w(m_k) \rfloor, \tag{7.27a}$$

$$\mu_{k+1} = [\mu_k + w(m_k)]_{\bmod 1}. \tag{7.27b}$$

Now define the function

$$\eta(m_k, i) = \mu_k + w(m_k) - i; \;\; i = 0, 1, 2, \ldots \tag{7.28}$$

At the basepoint m_k the value $\eta(m_k, 0)$ is stored in a b-bit register. At every $\mathrm{T_S}$ cycle the value of the register is decremented by 1,

$$\eta(m_k, i + 1) = \eta(m_k, i) - 1. \tag{7.29}$$

As long as $\eta(m_k, i) > 1$, there obviously exists an integer $\lfloor \mu_k + w(m_k) \rfloor > m_k + i$. The criterion to obtain the next basepoint m_{k+1} is $\eta(m_k, i_\mathrm{min}) < 1$, where i_min is the smallest integer for which the condition is fulfilled. Thus, the decrease operation is continued until the condition $\eta(m_k, i_\mathrm{min}) < 1$ is detected. By definition, the register content $\eta(m_k, i_\mathrm{min})$ equals μ_{k+1}. Afterwards, the operations are continued for m_{k+1} with the initial value

$$\eta(m_{k+1}, 0) = \eta(m_k, i_\mathrm{min}) + w(m_{k+1}). \tag{7.30}$$

Under an implementation point of view, the nominal ratio $\mathrm{T_I}/\mathrm{T_S}$ should always be larger than one to guarantee that no two basepoints coincide, i.e. $m_{k+1} \neq m_k$, in the noiseless case. Nevertheless, in our considerations the ratio $\mathrm{T_I} / \mathrm{T_S} = 1 + \chi$, where χ may be positive or negative small real value. Hence, the initialization carried out in (7.30) should be done with a modified control word $w'(m_{k+1}) = w(m_{k+1}) + \chi_\mathrm{max}$, where χ_max is a design parameter. Interestingly, simulation results for the case of the IEEE 802.11a standard have shown that by setting up the control word to $w(-1) = 1 - \chi_\mathrm{max}$ upon initialization of the timing loop and by considering the parameter K_w as in (7.24), it is always accomplished that $m_{k+1} = m_k + 1$ for any $|\chi_\mathrm{max}|<1$, so no variable decimator is necessary at all even with $\chi_\mathrm{max}=0$.

7.5 Simulation Results

7.5.1 Simplified RPE Estimator

The simplified RPE estimator in §7.3 (see Figure 7.1) has been initially simulated in an AWGN scenario using four different transmission rates: 9, 18, 36 and 54 Mbps as defined in the IEEE 802.11a standard. The SNR was selected to be 20 dB for the 9 and 18 Mbps cases, 30 dB for the 36 Mbps case and 40 dB for the 54

Mbps case. Furthermore, a residual carrier frequency offset after synchronization of 0.4%, i.e. 1.25 kHz, was forced in the 9, 18 and 36 Mbps cases. Due to its higher sensitivity, the 54 Mbps case was simulated with only 0.08% frequency error, i.e. 250 Hz. A sampling clock offset of $\zeta = -80$ ppm was selected together with a phase noise model as given in §3.5. No frame timing error was considered. According to the results shown in (3.24), the parameter δ in (7.7) is given by

$$\delta = \mathrm{L} \cdot \xi = \mathrm{L} \cdot 2\pi \cdot (4/3.2) \cdot \zeta, \tag{7.31}$$

being 4/3.2 the ratio between the symbol lengths with and without guard interval.

The TBL in the Viterbi decoder was selected to be 120 bits in all the cases. Furthermore, the SIGNAL symbol defined in [IEEE] is not intended for channel estimation but just for initialization. Therefore, according to the number of data bits per OFDM symbol given in Table 2.2, this results in the following values for L: L = 5 for 9 Mbps, L = 3 for 18 Mbps, L = 2 for 36 and 54 Mbps. The worst case, i.e. L = 5, results in $\delta = -5\cdot10^{-3}$ and the maximum $|\delta\cdot k| = 5\cdot10^{-3}\cdot26 = 0.13 << 1$, meaning that the simplification in (7.8) can be applied. The frame length was set to 58 data symbols in all the cases. The simulation results are shown graphically in Figure 7.5, where the constellation diagrams before (gray) and after (black) the phase correction are represented.

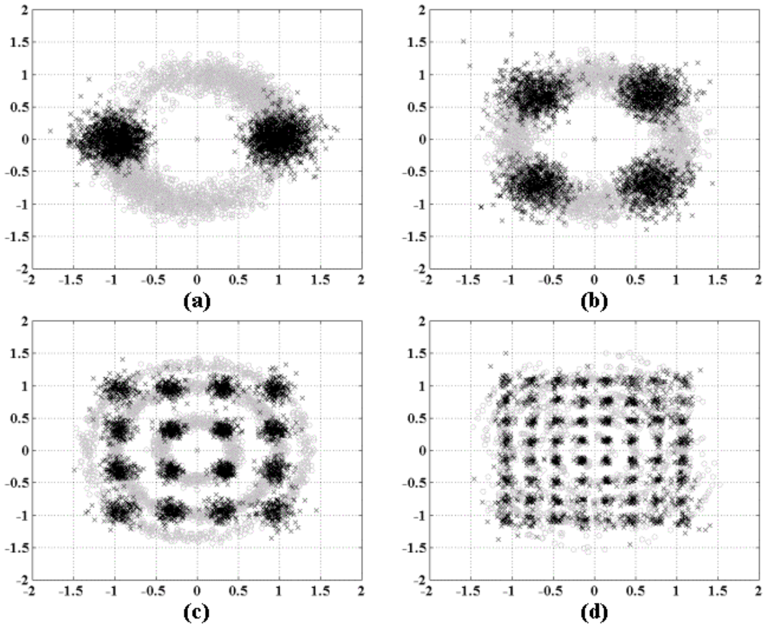


Figure 7.5. Simulated constellation diagrams before and after RPE correction for: a) 9 Mbps; b) 18 Mbps; c) 36 Mbps; d) 54 Mbps.

The results in Figure 7.5 have been shown to provide the reader a visual impression of what the phase corrector is doing. In order to evaluate analytically the performance of the RPE estimation and correction algorithm, a new figure of merit is introduced. The Error Vector Magnitude (EVM) is defined for a power-normalized constellation diagram as given in [IEEE],

$$\mathrm{EVM} = \sqrt{\frac{\sum_{l=1}^{\mathrm{L_S}}\sum_{k=1}^{48}\left(\Re\mathrm{e}\{\mathrm{e_C}(k,l)\}\right)^2 + \left(\Im\mathrm{m}\{\mathrm{e_C}(k,l)\}\right)^2}{48\cdot \mathrm{L_S}}}\ , \tag{7.32}$$

where L_S stands for the number of symbols contained in the frame and $e_C(k,l)$ is the constellation error given by

$$e_C(k,l) = A_{k,l} - \tilde{A}_{k,l} . \tag{7.33}$$

In (7.32) only the data sub-carriers, i.e. 48 out of 64, are considered in the calculation of the EVM. Simulation results are depicted in Figures 7.6/7.7 for the 9 Mbps and 54 Mbps cases, respectively. The results compare the EVM obtained versus the SNR for both AWGN and a fading channel according to the model A in Table D.1. For that, groups of 20 frames each containing 16 random symbols were generated. The sampling clock error was –80 ppm in all the cases. The residual carrier frequency offset was 0.4% (9 Mbps) and 0.08% (54 Mbps). The TBL was fixed to 120 bits.

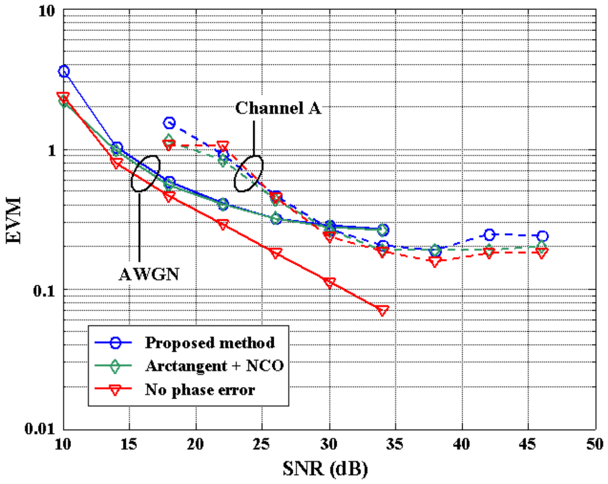


Figure 7.6. EVM versus SNR for the 9 Mbps case when the proposed RPE estimation and correction mechanism is used.

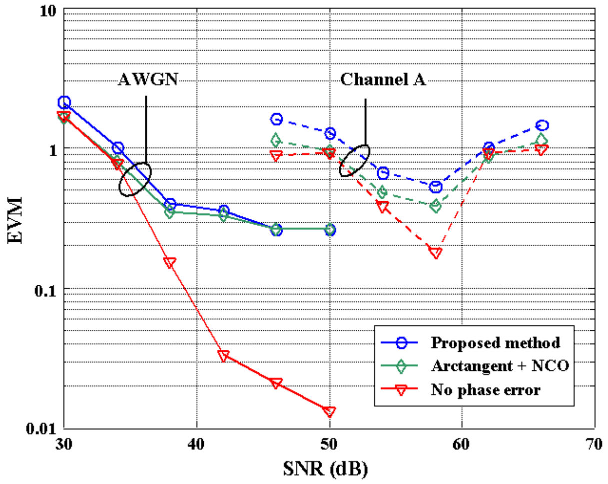


Figure 7.7. EVM versus SNR for the 54 Mbps case when the proposed RPE estimation and correction mechanism is used.

Figures 7.6 and 7.7 show that the proposed mechanism does not differ much from the ideal solution based on an Arctangent plus NCO in the AWGN case. In a multipath scenario, the proposed method suffers some performance degradation due to the CTF estimation errors. Nevertheless, a “floor” effect can be observed in both cases. The reason for this effect is the uncorrected timing errors. As shown in Chapter 3, when analyzing the impact of impairments on OFDM signals, timing and frequency errors not only appear as a linear phase after FFT but also introduce a certain amount of noise due to ISI and ICI. In this particular scenario the remaining timing errors play a key role and generate an interference power, which is above the channel noise power.

A last comment regarding the previous results is that apparently the simulations show a minimum value for EVM at a certain SNR when the Channel A is used. This is due to the fact that the simulations where carried out considering a time-direction filter in the CD3 channel estimator, as given in §5.3.3 (not shown in Figure 7.1), with a forgetting factor $\kappa = 0.001$, which was selected arbitrarily small to avoid disturbance with the phase estimator. It seems that these minima correspond to the SNR values for which the selected value of κ is the optimum one.

7.5.2 Timing Correction through Variable Interpolation

Simulations have been carried out in order to analyze the performance of the digital timing loop described in §§4.3.4 and 7.4. As an attempt to reduce the complexity of the simulation model, it is considered that the CD3 channel estimator is able to provide the correct estimation for the constellation points, i.e. $\hat{A}_{k,l-\mathrm{L}} \equiv A_{k,l-\mathrm{L}}$ (see Figure 7.1). Simulations are carried out for the rates 9 and 54 Mbps with the following loop delays: L= 5 (9 Mbps) and L=2 (54 Mbps). Each frame in the simulation has been randomly generated and contains 152 symbols, which is the longest possible frame length in the 54 Mbps case. Only a 1st order Farrow interpolator has been used in the simulations. The 2nd order Farrow interpolator leads to instabilities in the timing loop when the number of symbols is too big (several tens). The reason for that is that the approximation $\tilde{H}_{k,l} \approx \tilde{H}_{k,l-\mathrm{L}}$, which is the basis of the proposed RPE estimator, can no longer be accomplished when the 2nd order interpolator in Figure 7.4b is used. Simulation results are depicted in Figures 7.8-7.13.

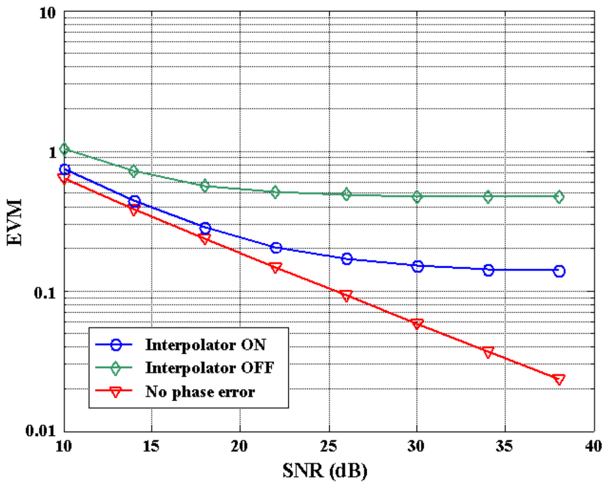


Figure 7.8. EVM versus SNR for the 9 Mbps case when the proposed variable time correction mechanism is used (AWGN).

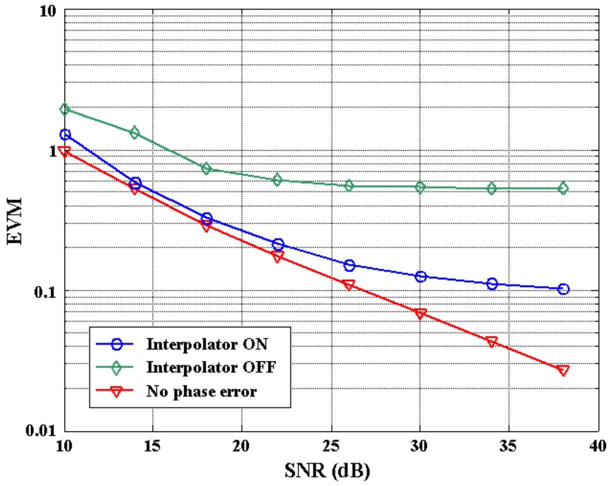


Figure 7.9. EVM versus SNR for the 54 Mbps case when the proposed variable time correction mechanism is used (AWGN).

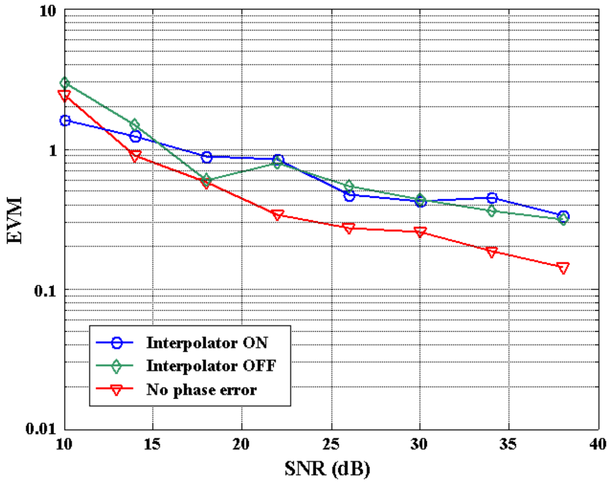


Figure 7.10. EVM versus SNR for the 9 Mbps case when the proposed variable time correction mechanism is used (Channel A w/o initial timing error).

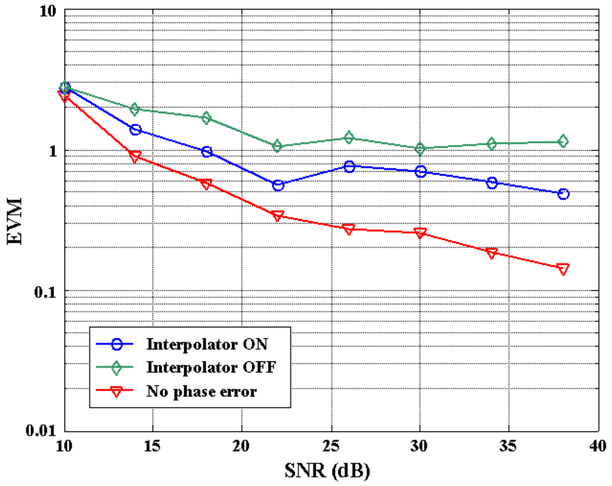


Figure 7.11. EVM versus SNR for the 9 Mbps case when the proposed variable time correction mechanism is used (Channel A w/ initial timing error = $0.5T_S$).

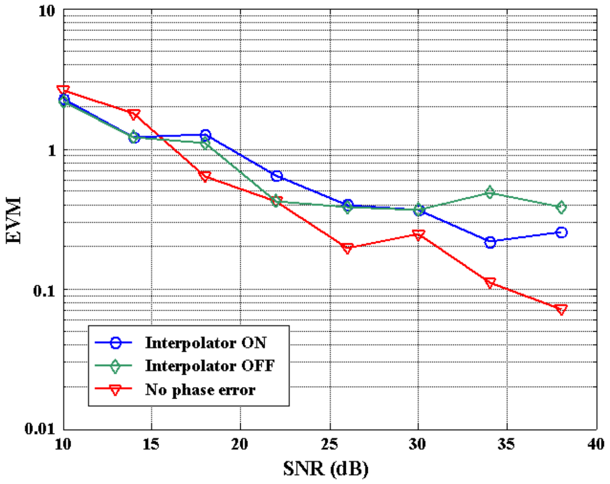


Figure 7.12. EVM versus SNR for the 54 Mbps case when the proposed variable time correction mechanism is used (Channel A w/o initial timing error).

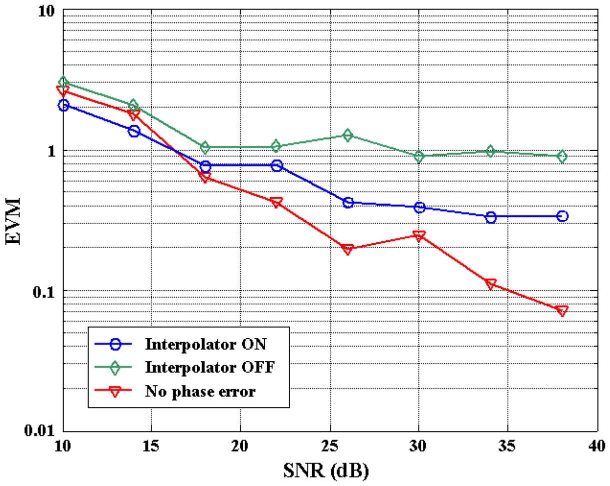


Figure 7.13. EVM versus SNR for the 54 Mbps case when the proposed variable time correction mechanism is used (Channel A w/ initial timing error = $0.5T_S$).

The results in Figures 7.8-7.9 show that the digital timing loop with 1st order Farrow interpolator provides significant improvement of the EVM in an AWGN scenario. In a fading channel scenario, no improvement is observed if only a sampling clock frequency offset is present (see Figures 7.10 and 7.12). Instead, the improvement is significant if an initial timing error derived from a possible timing synchronization error is present, as shown in Figures 7.11 and 7.13.

7.6 Architecture of System Components

In this section, the implementation of the most important conforming blocks in the Inner Receiver is addressed. Therefore, the selected architectures for FFT processor, complex dividers as well as the demapper and Viterbi decoder are presented. The final realization in form of an Application Specific Integrated Circuit

(ASIC) will be treated in Chapter 8 and should serve as a benchmark for comparison with other available solutions.

7.6.1 Details on the FFT Processor

The implementation of the FFT processor is based on the TURBO64 processor given by Maharatna in [Maha03a]. The 802.11a standard specifies that the 64-point FFT/IFFT has to be computed in 3.2 μs. The conventional Cooley-Tukey butterfly algorithm [Cool65] requires $(N/2)\cdot\log_2(N)$ complex multiplication operations. In our case, i.e. N = 64, this results in 192 complex multiplications, each being performed in 16.6 ns. Considering that a complex multiplication can be realized in one clock cycle, this results in an operating frequency of 60 MHz. Hence, it would be desirable to operate the targeted processor at 20 MHz frequency, since this is the timing of the input data. However, to satisfy the timing constraint at that frequency one needs to deploy several complex multipliers in parallel, which finally increases the silicon area and power consumption.

The FFT $X(k)$ of a complex data sequence $x(n)$ of length N, where $0 \le k, n \le \mathrm{N}-1$, can be described as

$$X(k) = \sum_{n=0}^{\mathrm{N}-1} x(n)\cdot W_{\mathrm{N}}^{n,k} \ , \tag{7.34}$$

with

$$W_{\mathrm{N}} = e^{-j\frac{2\pi}{\mathrm{N}}} \ . \tag{7.35}$$

According to [Maha03a], one may formulate the radix-8 representation of the 64-point FFT considering the following expansions of the variables k and n,

$$k = s + 8 \cdot t, \tag{7.36a}$$

$$n = l + 8 \cdot m, \tag{7.36b}$$

where $0 \le s, l, m, t \le 7$. Substituting these values into (7.34) and simplifying yields

$$X(s+8t) = \sum_{l=0}^{7}\left(W_{64}^{l,s} \sum_{m=0}^{7} x(l+8m)\cdot W_{8}^{m,s} \right)\cdot W_{8}^{l,t} \ . \tag{7.37}$$

Equation (7.37) shows that the 64-point FFT can be computed by performing two 8-point FFTs. The result of the first FFT has to be multiplied by the corresponding inter-dimensional coefficients before computation of the second FFT.

There are two important points to be noted in this implementation. On one hand, the realization of an 8-point FFT using the butterfly algorithm requires no explicit multiplication. On the other side, from the 64 inter-dimensional coefficients found in (7.37), i.e. $W_{64}^{l,s}$, only 49 are non-trivial. As a comparison, the radix-2 butterfly FFT requires 66 non-trivial multiplications.

The inverse FFT (IFFT) can be performed by first swapping the real and imaginary parts of the incoming data, then performing the forward FFT on them, and finally swapping the result once again at the output. A block diagram of the whole architecture is shown in Figure 7.14. With this architecture the parallel-to-parallel FFT computation requires 1.15 μs, i.e. 23 cycles @ 20 MHz.

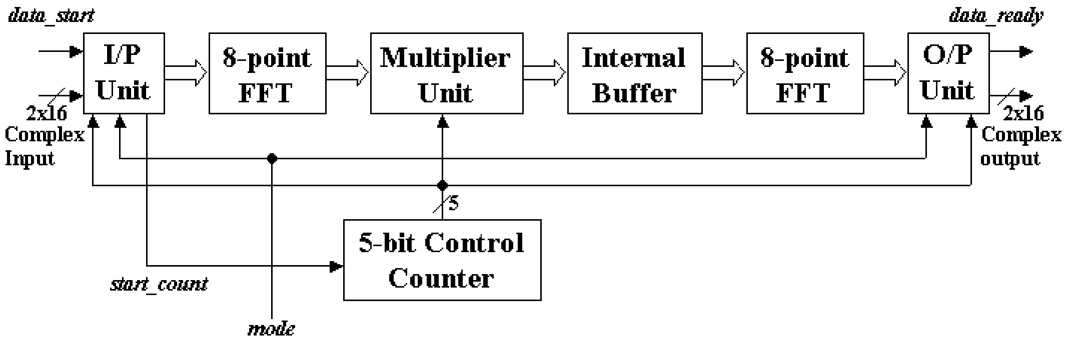


Figure 7.14. The architecture of the TURBO64 FFT/IFFT processor.

7.6.2 Design of the Complex Dividers

The main conforming blocks of the modified CD3 channel estimator have been shown in Figure 7.1. Special attention was given to the implementation of the RPE correction in §7.3. As it can be seen from the analysis done in §7.3, the *Equalizer* plays a key role in the simplification of the whole RPE estimation and correction mechanism.

The division of two complex values $a = a_\mathrm{R} + j\cdot a_\mathrm{I}$ and $b = b_\mathrm{R} + j\cdot b_\mathrm{I}$ results in $c = c_\mathrm{R} + j\cdot c_\mathrm{I}$ with

$$c = \frac{a_\mathrm{R} + ja_\mathrm{I}}{b_\mathrm{R} + jb_\mathrm{I}} \times \frac{b_\mathrm{R} - jb_\mathrm{I}}{b_\mathrm{R} - jb_\mathrm{I}} = \underbrace{\frac{(a_\mathrm{R}b_\mathrm{R} + a_\mathrm{I}b_\mathrm{I})}{|b|^2}}_{c_\mathrm{R}} + j\underbrace{\frac{(a_\mathrm{I}b_\mathrm{R} - a_\mathrm{R}b_\mathrm{I})}{|b|^2}}_{c_\mathrm{I}}. \tag{7.38}$$

Nevertheless, the following simplification of (7.38) is possible,

$$\begin{aligned} c &= \frac{1}{|b|^2}\big((a_\mathrm{R}b_\mathrm{R} + a_\mathrm{I}b_\mathrm{I}) + j(a_\mathrm{I}b_\mathrm{R} - a_\mathrm{R}b_\mathrm{I})\big) \\ &= \frac{1}{|b|^2}\big((a_\mathrm{R}b_\mathrm{R} + a_\mathrm{I}b_\mathrm{I} + a_\mathrm{R}b_\mathrm{I} - a_\mathrm{R}b_\mathrm{I}) + j(a_\mathrm{I}b_\mathrm{R} - a_\mathrm{R}b_\mathrm{I} + a_\mathrm{R}b_\mathrm{R} - a_\mathrm{R}b_\mathrm{R})\big) \\ &= \frac{1}{|b|^2}\big((a_\mathrm{R}(b_\mathrm{R} - b_\mathrm{I}) + b_\mathrm{I}(a_\mathrm{I} + a_\mathrm{R})) + j(a_\mathrm{R}(b_\mathrm{R} - b_\mathrm{I}) + b_\mathrm{R}(a_\mathrm{I} - a_\mathrm{R}))\big). \end{aligned} \tag{7.39}$$

In (7.39) the values c_R and c_I have a common element $a_\mathrm{R}\cdot(b_\mathrm{R} - b_\mathrm{I})$, which has to be calculated only once. Comparing (7.38) with (7.39) it comes out that the latter requires three more addition operations than the former, but at the same time it saves one multiplication operation, which comparatively is much more computational intensive than the three adders. In any case, two division operations are to be performed as explained next.

A *binary division algorithm* as given in [Pir98] is firstly analyzed here for a fixed-point implementation. In the division of A by D, the final result is given in a quotient part Q and a remainder part R as follows,

$$\frac{A}{D} = Q + \frac{R}{D}, \tag{7.40a}$$

or equivalently

$$A = Q \cdot D + R\,. \tag{7.40b}$$

For fixed-point division, the range of the operands must be predetermined. If Q and D are b-bit binary numbers, then A requires $2b$ bits and R requires b bits since $R < D$ must hold.
Considering that the numbers are positive integers, the quotient Q is given by

$$Q = \sum_{i=0}^{b-1} q_i\, 2^i\,. \tag{7.41}$$

Substitution of (7.41) into (7.40b) yields

$$A = q_{b-1} D 2^{b-1} + q_{b-2} D 2^{b-2} + \ldots + q_0 D 2^0 + R\,. \tag{7.42}$$

The evaluation of the quotient bits q_i is carried out iteratively from the highest valued bit q_{b-1} to the lowest valued bit q_0. Once a particular quotient bit has been calculated, its corresponding term is subtracted from the value of A and a sequence of positive remainders R_j is created. The algorithm starts with the initialization

$$R_0\, 2^b = A\,. \tag{7.43}$$

Now using (7.42), a recursive equation for the remainder can be formulated,

$$R_j\, 2^{b-j} = q_{b-(j+1)} D 2^{b-(j+1)} + R_{j+1} 2^{b-(j+1)}; \quad j = 0,1,\ldots,b-1\,. \tag{7.44}$$

The recursive algorithm for the evaluation of the quotient bits can now be given. The first step is

$$R_0 = A 2^{-b}\,, \tag{7.45}$$

which means that the point in A is shifted by b digits. By extracting the common factor 2^{b-j} in (7.44), the new remainder results from the previous one as follows,

$$R_{j+1} = 2R_j - q_{b-(j+1)} D\,; \quad j = 0,1,\ldots,b-1\,. \tag{7.46}$$

Since the goal is to create a sequence of smallest possible remainders, i.e. $R_0 > R_1 > \ldots > R_{b-1}$, the bits $q_{b-(j+1)}$ are selected as follows,

$$q_{b-(j+1)} = \begin{cases} 0\,; & 2R_j < D \\ 1\,; & 2R_j \geq D \end{cases}. \tag{7.47}$$

Nevertheless, the previous solution is not applicable if a pipeline-type architecture is envisioned. Such an approach is specially interesting since pipelines introduce an initial delay in the calculation but afterwards they provide a result at each clock cycle. The solution we propose is based on the linear CORDIC working in the *vectoring mode*. For that we make use of expression (A.1) in Appendix A, which embodies any coordinate rotation in three different modes of representation, namely circular, linear and hyperbolic. We will restrict ourselves to the case m → 0 (linear representation), resulting in the following rotation matrix R(θ),

$$\begin{pmatrix} x_{\text{out}} \\ y_{\text{out}} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ -\theta & 1 \end{pmatrix}}_{\mathrm{R}(\theta)} \begin{pmatrix} x_{\text{in}} \\ y_{\text{in}} \end{pmatrix}. \tag{7.48}$$

This results in the following equations relating $(x_{\text{in}}\ y_{\text{in}})^{\mathrm{T}}$ and $(x_{\text{out}}\ y_{\text{out}})^{\mathrm{T}}$,

$$x_{\text{out}} = x_{\text{in}}\,, \tag{7.49a}$$

$$y_{\text{out}} = y_{\text{in}} - \theta \cdot x_{\text{in}} \,. \tag{7.49b}$$

In the vectoring mode of operation, the CORDIC processor rotates $(x_{\text{in}}\ y_{\text{in}})^{T}$ until an output vector with $y_{\text{out}} = 0$ and $x_{\text{out}} = x_{\text{in}}$ is obtained (see Figure 7.15). A variable z, which is initialized to 0 and is only considered for implementation, accumulates all the rotations needed to achieve this output vector. Thus, (7.49b) results as follows,

$$\theta = \frac{y_{\text{in}}}{x_{\text{in}}} \,, \tag{7.50}$$

and the variable z_{out} contains θ, i.e. the final result of the division.

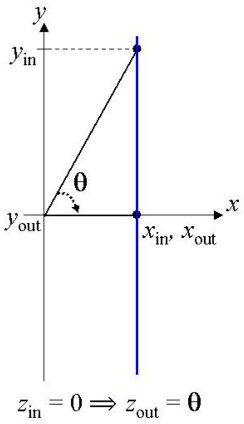


Figure 7.15. Linear CORDIC operated in the vectoring mode.

Generally, the vector θ is expressed as an addition of b elementary rotations, i.e. $\theta = \alpha_0 + \alpha_1 + \ldots + \alpha_{b-1}$. Therefore (7.48) may be expressed as

$$\begin{pmatrix} x_{\text{out}} \\ y_{\text{out}} \end{pmatrix} = \prod_{i=0}^{b-1} \begin{pmatrix} 1 & 0 \\ -\alpha_i & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{\text{in}} \\ y_{\text{in}} \end{pmatrix} . \tag{7.51}$$

Each stage i will have three input signals (x_i, y_i, z_i) and three output signals $(x_{i+1}, y_{i+1}, z_{i+1})$ obtained as follows,

$$x_{i+1} = x_i \tag{7.52a}$$

$$y_{i+1} = y_i - \alpha_i x_i \tag{7.52b}$$

$$z_{i+1} = z_i + \alpha_i \,. \tag{7.52c}$$

Furthermore, the elementary rotations are given by the following expression,

$$\alpha_i = 2^{b-1-i} \,. \tag{7.53}$$

In our implementation only positive values x_{in} and y_{in} are considered. The first stage ($i = 0$) is initialized with $x_{\text{in}} = D$ (16 bit), $y_{\text{in}} = A$ (32 bit), $z_{\text{in}} = 0$ (16 bit). At each stage i, the input x_i is right-shifted b–1–i positions. Afterwards, the operation in (7.52b) will only be performed if

$$y_i > 2^{b-1-i} x_i \,,$$

and if the b–1–i MSBs in x_i are all zero. In this case, (7.52c) is equivalent to switching the bit at position b–i in z_i to a logic '1'. If both conditions are not satisfied, then $y_{i+1} = y_i$ and $z_{i+1} = z_i$. Note that this solution is equivalent to the *pen-and-paper* method of solving divisions.

The complex divider conforming the equalizer shows two main drawbacks. In one hand, it enhances the CTF estimation noise; on the other side, it is a computationally intensive block that requires a relatively large number of bits in order to reduce the induced digital noise. The authors in [Fech99] propose a solution in which the equalization of the input sequence $Y_{k,l}$ (see Figure 7.1) is performed in two steps, avoiding any division operation. The first step, referred as phase compensation, complies the multiplication of $Y_{k,l}$ by the sequence $\tilde{H}^{*}_{k,l}$, being $\tilde{H}_{k,l}$ an estimation of the CTF based on the pilot tones in symbol *l*. A scale factor sequence of the form $|\tilde{H}_{k,l}|^2$ still remains after multiplication, being eliminated in a second step by the demapper (scale factor compensation). The demapper makes use of the sequence $|\tilde{H}_{k,l}|^2$ as an input signal to accordingly scale the constellation diagram prior to the assignment of the soft-bits. Nevertheless, this solution is only valid if the pilot tones carried into a particular OFDM symbol are used to obtain the CTF estimation for that symbol. Last but not least, the design of a scaling demapper is not straightforward. In [Fech99] an iterative algorithm is proposed, which increases the overall latency and therefore it is not suited for wireless LAN systems, where latency is a key parameter.

7.6.3 Demapper and Decoder Blocks

The blocks placed after the phase correction mechanism in Figure 7.1 correspond to the *demapper*, *deinterleaver* and the *Viterbi decoder*, which should extract the correct bits from a noisy constellation diagram. Two strategies, namely hard- and soft-decision decoding, are possible as explained in [Pro95]. If a hard-decision strategy were selected, the demapper would generate an estimation of the bits based on the Euclidean distance. Considering the constellation point marked with an 'x' in Figure 7.16 (without normalization), the “hard” demapper would decide the bits [1 1 1 1] as the most likely received bits. After deinterleaving, the Viterbi decoder should correct those bits, which were falsely decided during demapping.

In the soft-decision approach, the demapper makes no direct decision on the bits themselves, but it generates an estimation of their *reliability*, whereas the final decision is left to the Viterbi decoder. The assignment of a reliability value is not trivial and is generally based on a non-linear rule, as analyzed in [Tos02]. The reliability is an integer value coded with a number of bits N_{soft}, ranging from $(-2^{Nsoft-1} + 1)$ to $2^{Nsoft-1}$, the most negative value representing a very reliable ‘0’, whereas the most positive one stands for the most reliable ‘1’. The least reliable ‘0’ and ‘1’ are the integer values 0 and 1, respectively.

In the case of the 802.11a and HL2 standards, the maximum constellation diagram is 64-QAM, i.e. each point on the constellation corresponds to 6 bits (named as I_0 I_1 I_2 Q_0 Q_1 Q_2) as shown in Figure 7.17. In order to decide the value of reliability to be assigned to each bit, we refer to Figure 7.17. As it can be seen, the bit I_0 is ‘1’ if the I component is positive and ‘0’ otherwise. The same applies to the bit Q_0 with respect to the Q component. Hence the reliability value for these two bits may be obtained as given in Figure 7.18a (for a normalized input). The function shown there is obviously piecewise discontinuous since a limited number of bits are used to represent the reliability value. In this implementation it is quantized with four bits ($N_{soft} = 4$), resulting in 16 possible regions. Similarly, the bits I_1 and Q_1 may be assigned a reliability value according to Figure 7.18b. As it can be seen from Figure 7.18b, the maximum reliability value for the bit I_1 (Q_1) is one half of the maximum corresponding to I_0 (Q_0). Finally, reliability values for I_2 and Q_2 result from the function depicted in Figure 7.18c. In this case, the maximum reliability that can be assigned to I_2 (Q_2) is one half of the maximum for I_1 (Q_1). This *weighted* reliability has its explanation on the fact that the bits I_2 (Q_2) are more affected by noise than I_1 (Q_1), and at the same time the bits I_1 (Q_1) are noisier than I_0 (Q_0). If I_2 (Q_2) may take the same reliability values as I_0 (Q_0), the noise would be amplified.

For the other modulation schemes being used, i.e. BPSK, QPSK and 16-QAM, the procedure can be derived directly from the previous result (for normalized constellation diagrams). Hence, in a BPSK scheme, only the reliability value obtained for I_0 is to be considered for further processing. In QPSK, the relevant values are I_0 and Q_0 whereas, in the 16-QAM case, I_0, I_1, Q_0 and Q_1 are the key bits.

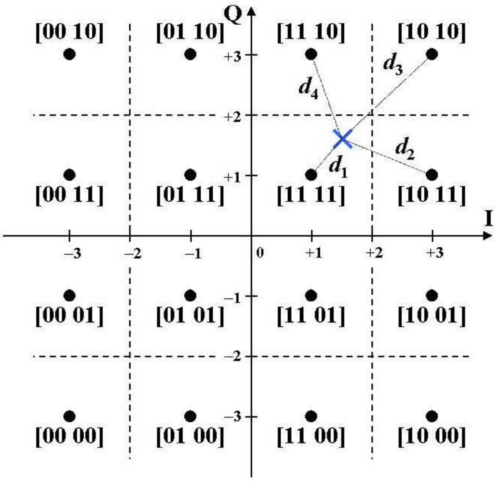


Figure 7.16. Decision bounds in a hard-decision demapper.

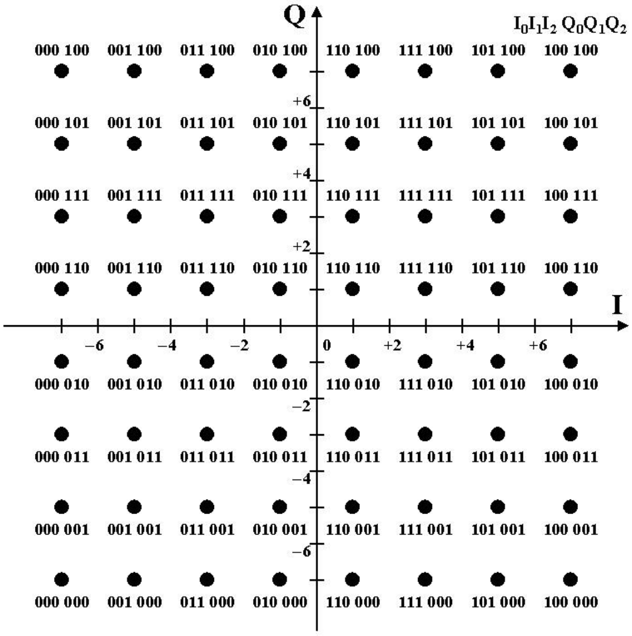


Figure 7.17. Constellation diagram for the 64-QAM modulation scheme.

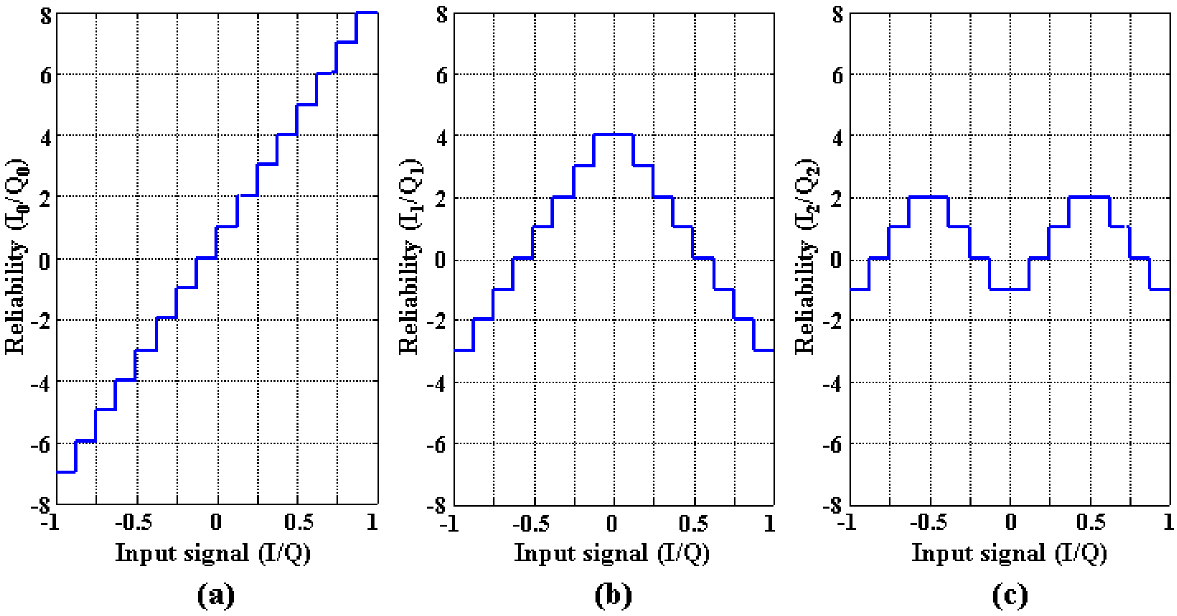


Figure 7.18. Reliability assignment ($N_{soft} = 4$) versus normalized inputs I/Q for: a) I_0 and Q_0; b) I_1 and Q_1; c) I_2 and Q_2.

The processing block coming next to the demapper is the deinterleaver, Figure 7.1. Its operation was already described in §2.4.1. Nevertheless, the *soft*-deinterleaver is not going to work with single bits, but with integer values, i.e. groups of bits representing a reliability estimation. Its implementation does not impose major challenges; nevertheless, since deinterleaving is intrinsically based on read and write operations into a memory, the design may result rather area consuming.

However, the most important block inside the whole Forward Error Correction (FEC) mechanism is the *Viterbi decoder*. The Viterbi decoder applies the Viterbi algorithm to decode a sequence of bits, which was previously encoded through a convolutional encoder [Vit67]. This method of decoding represents the Maximum Likelihood estimation of the transmitted sequence of bits and thus, it is asymptotically optimal.

Prior to the decoding process itself, a step of “bit insertion” (or “depuncturing”) is required in those transmission rates with a code rate different of 1/2. This is necessary to compensate the puncturing introduced at the transmitter side (see §2.4). The way these *virtual* bits are introduced is by alternatively assigning a least confident (or reliable) ‘1’ or ‘0’ bit, i.e. the integer values 1 and 0. In order to clearly distinguish the bits I_2 (Q_2) from the punctured bits, the maximum reliability to be assigned to I_2 (Q_2) should always be higher than the reliability assigned to the punctured bits. This is achieved by employing at least four soft-bits, as it is the case in Figure 7.18. Logically, the punctured bits are less reliable than the bits I_2 (Q_2) since they are the result of a random assignment.

Two main parameters define the Viterbi decoder, namely N_{soft} and TBL. Regarding the number of soft bits, computer simulations have shown that a value larger than three ($N_{soft} = 3$) results in very little improvement of the decoding process [Pro95]. On the other side, for traceback lengths higher than 5K bits, where K is the length of the convolutional encoder (K=7 in the 802.11a and HL2 standards), also little improvement in performance is achieved. Nevertheless, these statements are only true when no puncturing is applied to the coded bits. In case of puncturing, we have already established that at least four soft-bits should be used when a 64-QAM modulation scheme is applied. Furthermore, some authors suggest using longer TBLs, in the range of 7K to 9K, in order to compensate the additional noise produced by the bit uncertainty [Hend96], [Flem03]. By computer simulations (see §5.3.5) we have established that a TBL of at least 50 bits is necessary.

The Viterbi decoder considered in our implementation makes use of four soft bits. Furthermore, the decoder applies two different traceback lengths according to the symbol being decoded. Hence, the SIGNAL symbol (see §2.4.2), which is always BPSK modulated, is decoded with 24 bits traceback length whereas, the DATA symbols are decoded with 48 bits. The algorithm introduces an initial delay and afterwards provides a full byte (8 bits) per clock cycle, which effectively reduces the traceback length to 40 bits. The Add-Compare-Select (ACS) block is operated at 80 MHz, whereas the traceback memory operates at 10 MHz. The selection of a TBL of 40 bits is mainly due to hardware restrictions. The present Design Kit used at the IHP does not include any model for RAM cells. In addition, no RAM generation tool is available. For that reason, the traceback memory in the Viterbi decoder had to be generated using normal registers, which resulted in area requirements in the order of four to five times higher than in a traditional RAM [Maha03b], e.g. a D Flip Flop requires 28 transistors whereas the normal memory cell requires only 6.

7.7 Summary

Decision-directed channel estimation is the most popular way to perform CTF estimation in an OFDM system with a small number of sub-carriers. Nevertheless, pilot sub-carriers cannot be completely suppressed when coherent modulation is used since they provide the means for RPE estimation. In the present Chapter, a novel method for estimation and correction of the RPE has been presented, which highly simplifies the common method based on Arctangent (for phase calculation) plus NCO (for phase correction).

The proposed RPE estimation method provides an error signal, which can be further used in the estimation of the sampling clock frequency offset. As it was shown in §3.4, an offset in the sampling clock frequency is equivalent to a variable timing offset. The symbol timing is corrected by a variable interpolator, which is based on Farrow structures due to their simple implementation. Simulation results show that only first order Farrow interpolators can assure stability of the DTL. The reason for this is that the main assumption made during definition of the simplified RPE estimator no longer holds when higher order interpolation filters are used. Nevertheless, the proposed DTL provides a big improvement of the EVM in an AWGN channel when a sampling clock error is present. Simulations were also carried out in a multipath scenario in two cases namely, when only a sampling clock error was present and when the sampling clock error was together with an initial symbol timing estimation error of half a sample. Results show that in the former case (sampling clock error only) no significant improvement of the EVM was achieved by the DTL. On the contrary, when an initial timing error is present, the improvement is significant. With this analysis, the elements involved in a DTL have been finally defined, thus completing the general scheme provided in §4.3.4.

The Chapter is finished with a description of some of the key blocks conforming the Inner Receiver. The 64-point FFT/IFFT processor has been simplified by converting it into a two-dimensional 8-by-8 FFT/IFFT operation. In this manner, the number of non-trivial complex multiplications has been reduced by 26%. Furthermore, the proposed FFT/IFFT processor is able to perform an operation in only 23 clock cycles (parallel-to-parallel). A second important block is the complex divider included in the Equalizer, which requires two parallel divisions. Each divider is based on the linear CORDIC algorithm working in the vectoring mode, thus enabling a simple pipeline structure. The last block considered in these lines is the Viterbi decoder. The proposed *soft* Viterbi decoder operates with four soft-bits and a TBL of 48 bits. Since a full byte is provided at each clock cycle, the effective TBL is reduced to 40 bits due to hardware restrictions.

Two other crucial blocks involved in the design of the Inner Receiver have not been treated in this Chapter namely, the NCO/Arctangent block and the Interpolation/Decimation filters. Their analysis is considered later on in Appendices A and B, respectively.

Chapter 8

VLSI Implementation of the Inner Receiver

8.1 Introduction

BASED ON THE STRUCTURES DERIVED IN CHAPTERS 6 AND 7 for the Synchronizer and the Channel Estimator, respectively, we now concentrate on the VLSI implementation of these architectures. Hence, §8.2 is initially devoted to the realization of a low-power version of the Synchronizer. The low-power performance not only is achieved by designing carefully each component in the Synchronizer, but also by dividing the whole architecture into different clock domains, where each domain is solely activated when required and disabled afterwards. The integration of the Synchronizer and the Channel Estimator onto a single chip is achieved after several attempts in §8.3. Three experimental versions of the BBP labeled as BBP1, BBP2 and BBP3 were generated, each one being an improved version of the previous one. The latest version BBP3 achieves power figures, which are as good as the figures reported by renowned manufacturers in this field like Atheros or Toshiba. The results presented in this Chapter have been partly reported by the author in [Krstic03a] and [Krstic03b].

8.2 Mechanisms for Power-saving inside the Synchronizer

8.3.1 Clock Domain Separation

The general structure of the proposed Synchronizer is shown in Figure 8.1. Two mutually exclusive operations are present in the system: tracking for the training sequences and processing of the observed symbols. This fact can be used to obtain a power efficient synchronizer design by applying clock gating. Hence, the whole structure has been split into three clock domains and two mutually exclusive paths: *tracking data path* and *processing data path*, Figure 8.1. The main function of the tracking data path is to detect an incoming frame by searching for the periodic structure of the preamble symbols and to estimate the carrier frequency offset. The activity of the processing data path starts when the frame is detected and the estimated value for ε is available. This part of the synchronizer performs the carrier frequency error correction (NCO), estimates the symbol timing (crosscorrelator) and obtains the reference channel estimation.

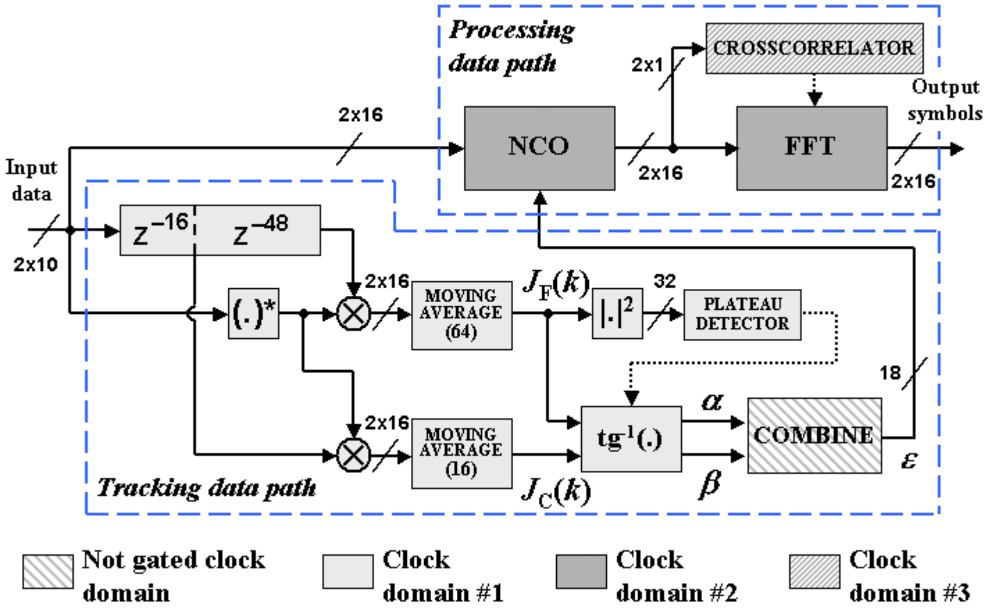


Figure 8.1. General block diagram of the proposed Synchronizer with clock domain separation.

The Arctangent operation and the NCO are realized using the CORDIC algorithm in the vectoring and rotational mode of operation, respectively. While the rotation mode of operation of the CORDIC enables us to compute the multiplication of any quantity with a phasor $\exp\{j\varphi\} = \cos(\varphi) + j\sin(\varphi)$, the vectoring mode can be used for computing the magnitude and the phase angle of a complex value. The operation principle of these two modes is shown in Figure 8.2. For the realization of the Synchronizer, the two modes of operation of the CORDIC are independently utilized in two different phases. At first, the phases $\angle J_F(k)$ and $\angle J_C(k)$ (see Figure 8.1) are computed (vectoring mode of operation) to obtain α and β, respectively. After calculation of ε by combination of α and β, this value is used to obtain a phase correction for the incoming data symbols applying a Numerically Controlled Oscillator (NCO) in the rotation mode of operation. The phase angle evaluation takes place only once at the beginning of the frame (after frame detection) whereas, for the rest of the frame the NCO operation is carried out. Thus, in our consideration, it is more pragmatic to separate the two modes of CORDIC operation and realize them in the form of two separate modules. Several benefits arise from this separation. On one hand, the control mechanism is greatly simplified compared to the use of a single full CORDIC. Furthermore, although the total silicon area is 33% more when two processors are used, the power consumption is reduced by 54%. When using the rotational CORDIC as an NCO, the input signal is multiplied by a phasor with the form $\exp\{j(2\pi/64)\cdot\varepsilon\cdot n\}$, where ε is the normalized carrier frequency and n is the time variable. Thus, the variable φ in Figure 8.2a depends on n, and it is updated at each clock cycle. This is achieved by adding a phase accumulator to the input φ in the CORDIC processor (Figure A.19). Note that the value of ε must not be sign-reversed in order to apply a correct phase correction, because the result coming from the Arctangent calculation already considers this sign. This can be seen in (6.6), where by definition the phase contribution in $J(k)$ already contains the *–ve* sign. The actual implementation of the rotational and vectoring CORDIC processors is described in more detail in Appendix A.

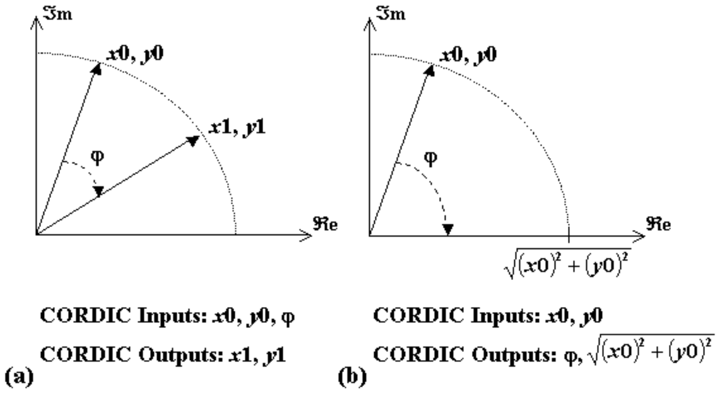


Figure 8.2. Principle of operation of the circular CORDIC algorithm: a) Rotation mode; b) Vectoring mode.

8.3.2 Synthesis Results

The proposed Synchronizer has been synthesized in IHP's owned 0.25 μm SiGe:C BiCMOS technology. After synthesis, the power analysis made by Synopsys resulted in power figures of 140 and 175 mW, with and without clock gating, respectively [Krstic03a]. The core layout area is 13 mm^2 and is shown in Figure 8.3. Power figures and cell area results are provided in Table 8.1 for the most relevant blocks found in the Synchronizer. The whole structure introduces a latency of 3.9 μs counted from the instant when the frame is detected. This value is less than one OFDM symbol period (4 μs), meaning that no additional storage of the input samples is necessary inside the Synchronizer.

Table 8.1. List of components of the Synchronizer.

Component	Cell area (mm^2)	Power (mW)
FFT	3.37	140*
NCO	0.78	6.6
Arctangent	0.8	7.2
64-tap Delay line	0.49	110
Moving Average (16 & 64)	0.49	115
Square magnitude	0.12	3.8
Plateau detector	0.23	8
α and β combining	0.06	0.8
Crosscorrelator	0.1	1

* Measured power consumption after fabrication as a discrete component is 84 mW @ 2.5 V, 20 MHz [Maha03a].

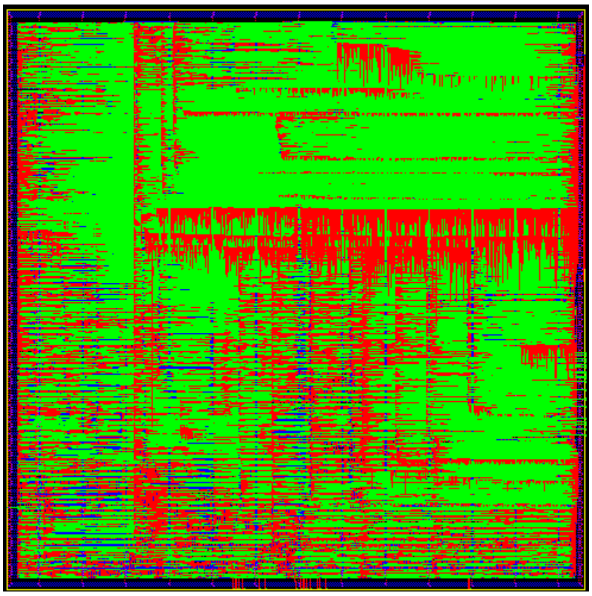


Figure 8.3. Core layout of the Synchronizer.

8.3 Implementation of the Baseband Processor

8.3.1 The Baseband Processor #1 (BBP1)

The BBP1 chip was initially developed to contain only the datapath as defined by the IEEE 802.11a standard. Thus, neither the Synchronizer nor the Channel Estimator where included. Apart from the datapath, the BBP1 includes an Enhanced Parallel Port (EPP) to provide an interface to the upper layers. This initial design was far away from being optimized and resulted in a silicon area of about 107 mm^2 (including pads), with 3.44 Million transistors and an estimated power consumption of 1 W. A block diagram is shown in Figure 8.4.

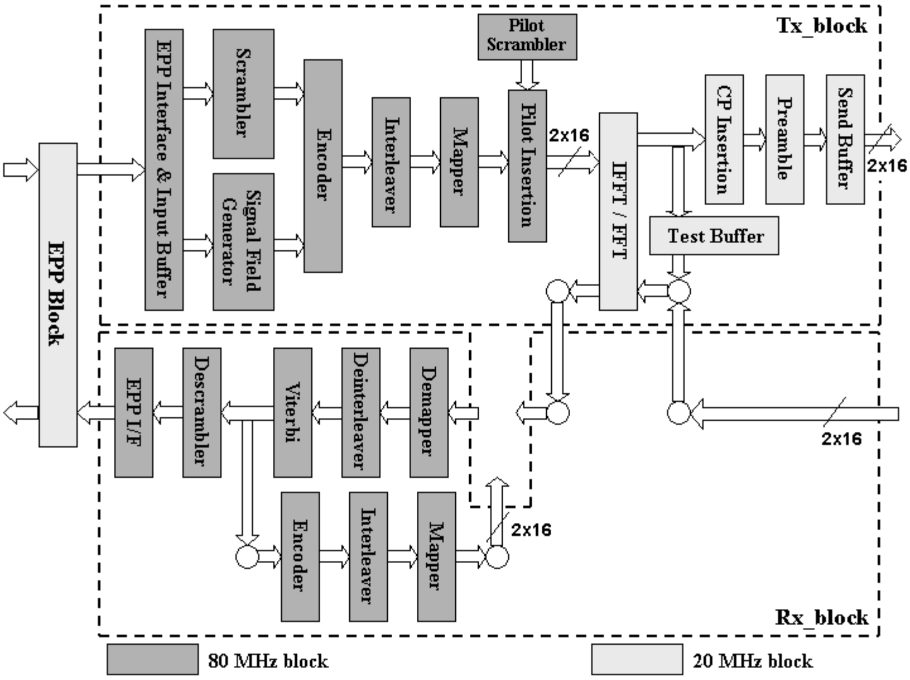


Figure 8.4. Block diagram of the BBP1.

The scheme shown in Figure 8.4 incorporates several buffers. One of these buffers is included inside the EPP block and is able to store up to 4 Kbytes, which corresponds to the maximum frame length in the 802.11a standard. A second buffer is called *Test Buffer* and is used for testing purposes. It can store up to four OFDM symbols, i.e. 2x16x4x64 = 1Kbyte, as is placed after the IFFT block. All these buffers were generated using D Flip-flops, with the subsequent increase in area (D Flip-flops require between four to five times more transistors than a normal memory cell). The final layout of the BBP1 is shown in Figure 8.5. Due to our lack of experience with the floorplanning tool, a significant amount of area was allocated only for wiring as it can be seen from Figure 8.5. The first electrical measurements showed that only 7% of the fabricated chips presented a standby current. This low *yield* value was mainly due to the large chip area, which increased the probability of fabrication errors.

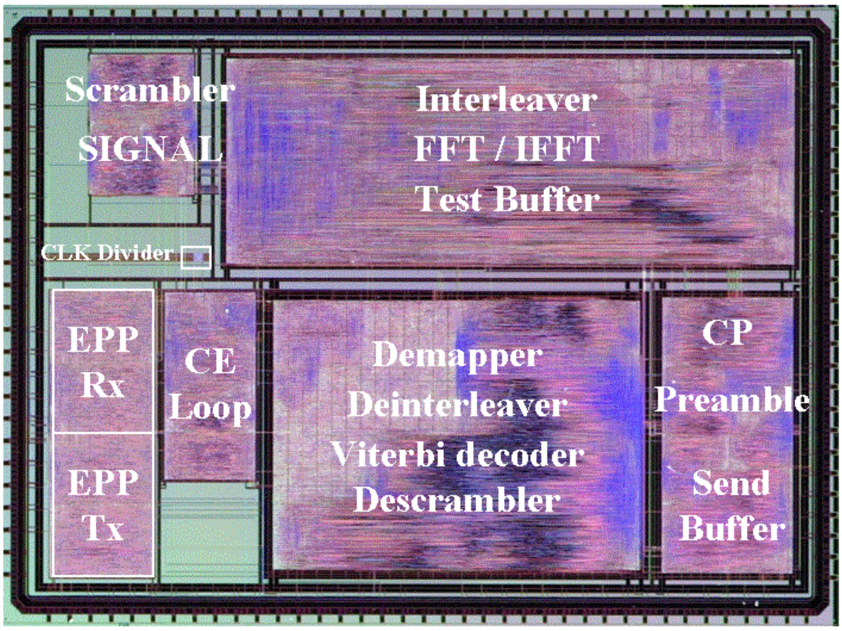


Figure 8.5. Layout of the BBP1.

8.3.2 The Baseband Processor #2 (BBP2)

In order to simplify the data processing and optimize the power consumption to a maximal extent, the structure of the Baseband processor was divided into three clock domains [Krstic03b]. Computationally complex blocks without high data throughput requirements were designed for 20 MHz, whereas high data throughput demanding circuits were designed for 80 MHz. The interpolation and decimation filters were designed following the results in Appendix B, which also require an intermediate step running at 40 MHz. A block diagram of the BBP2 is shown in Figure 8.6. Apart form the standardized Tx and Rx blocks, the Baseband processor includes the Synchronizer scheme given in §8.2.

Nevertheless, two main components have not been included in this experimental Baseband processor. In one hand, no circuitry for timing correction is included, thus limiting the maximum number of symbols per frame that can be transmitted. Furthermore, the noise reduction matrix conforming the Frequency Direction Filter in Figure 7.1 is also not included due to its computational complexity. As a result, the throughput for the 48 and

54 Mbps transmission modes will be drastically reduced. An optimum way to implement the NRM operation is still under investigation.

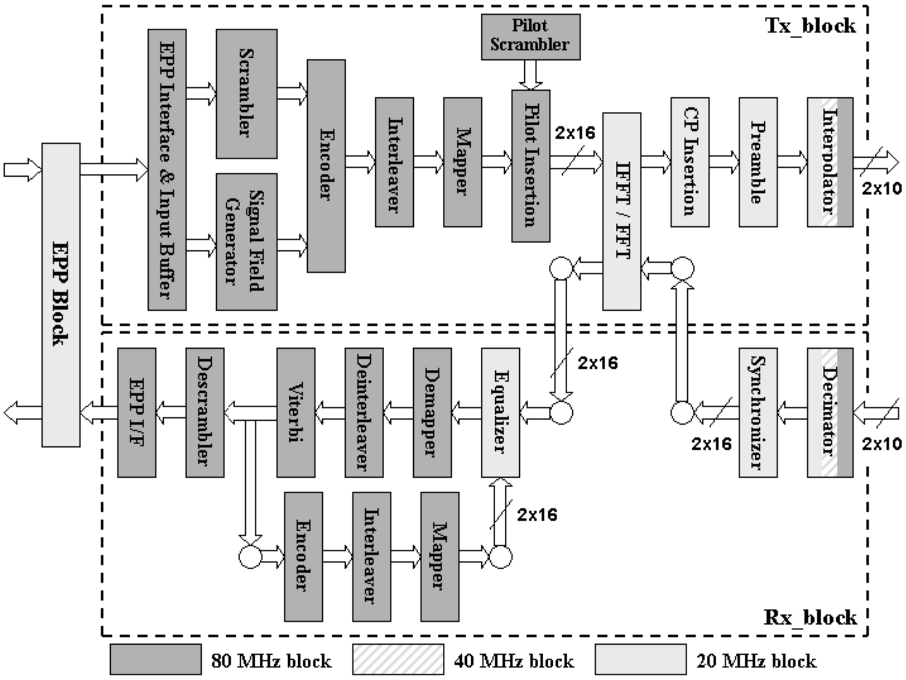


Figure 8.6. Block diagram of the BBP2.

The complete Baseband processor was modeled in VHDL and synthesized using the IHP's 0.25 µm 5-metal layer SiGe:C BiCMOS standard cell library. Its cell area, including all the blocks shown in Figure 8.6, is 24 mm^2 (equivalent transistor count is 1.55 Million), which represents a significant reduction with respect to the BBP1. In Table 8.2 some synthesis results for this technology are given, where Tx_* indicates transmitter components and Rx_* stands for the receiver components.

From Table 8.2 it can be noticed that the dominant hardware part is the receiver. It occupies 52% of the chip cell area, with the most silicon consuming components being the Synchronizer, the Channel Estimator, the Viterbi decoder and the FFT/IFFT processor. According to post-synthesis power estimation obtained by the *Synopsys Digital Compiler* tool, the expected power consumption is 860 mW in the receive direction and 795 mW in transmit direction. Nevertheless, the actual power is expected to be lower than that due to the pessimistic nature of the power estimations realized by this CAD tool.

The layout of the Baseband processor is given in Figure 8.7. This chip has been floorplanned into four main layout blocks of about equal size: EPP (6.7 mm^2), TX (Transmitter, 11 mm^2), RX_1 (Receiver, 20 MHz block, 9 mm^2) and RX_2 (Receiver, 80 MHz block, 10.4 mm^2). Additionally, there is one small block that performs clock division and clock distribution. After layout, the silicon area is 46 mm^2 if only the core is considered, and 59 mm^2 if the pads are included. The number of pins is 107. Figure 8.8 shows the die photograph of the fabricated chip, which is currently under test.

A significant improvement in area and power consumption was achieved in the BBP2 in comparison with the BBP1. Nevertheless, the layout in Figure 8.7 shows that still a significant amount of silicon area was still only used for routing. In order to avoid this, a re-design was mandatory.

Table 8.2. Synthesis results obtained for the BBP2.

Hardware component		*Cell area (mm^2)*		*Percentage of occupied area*	
Tx		**2.7**		**11.3**	
	Tx_interleaver		0.55		2.3
	Tx_pilot_insertion		0.66		2.7
	Tx_CP_insertion		0.5		2.0
	Tx_input_buffer		0.47		1.9
	Tx_interpolator		0.32		1.3
Rx		**12.5**		**52.0**	
	Rx_synchronizer		3		12.5
	Rx_channel_estimator		2.76		11.5
	Rx_viterbi_decoder		3.53		14.7
	Rx_channel_est_buffer		0.79		3.3
	Rx_deinterleaver		1.0		4.2
	Rx_demapper		0.29		1.2
	Rx_interleaver		0.55		2.3
	Rx_decimator		0.33		1.4
FFT / IFFT		**4**		**16.7**	
EPP		**4.8**		**20.0**	
Total		**24**		**100.0**	

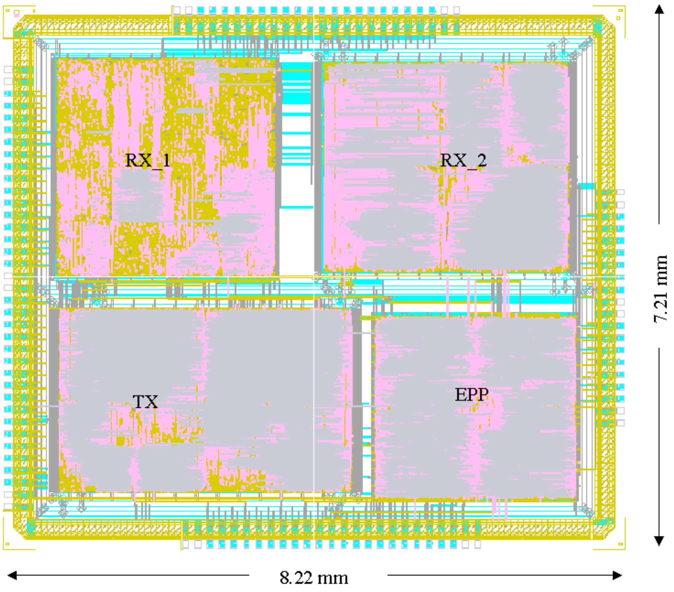


Figure 8.7. Layout of the BBP2.

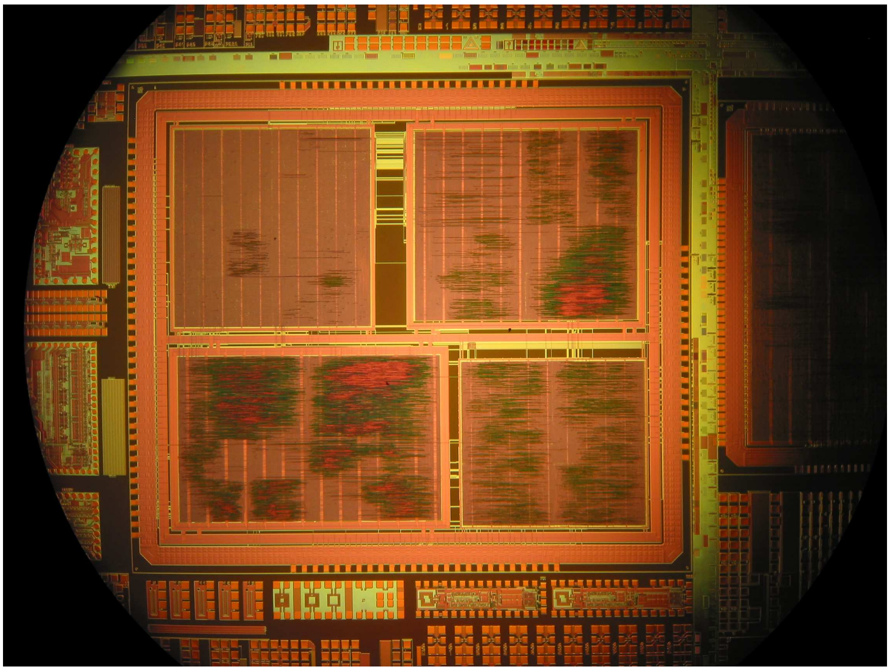


Figure 8.8. Die photograph of the BBP2.

8.3.3 The Baseband Processor #3 (BBP3)

In order to simplify the testing of the chip, both the EPP and the Interpolation/Decimation blocks were excluded from the BBP2. The resulting block diagram of the BBP3 chip is sketched in Figure 8.9. This new design results in a synthesized area of approximately 34 mm^2, with a core area of only 19.5 mm^2 and 124 pins, see Figure 8.10.

Table 8.3. Synthesis results obtained for the BBP3.

Hardware component	*Average power (mW)*
Tx Core	**104**
FFT / IFFT	40.6
Interleaver	13.1
Pilot Insertion	31.8
Others	18.5
Rx Core	**146**
Synchronizer	25.7
Channel Estimator	15.4
Viterbi Decoder	53.7
Deinterleaver	20.6
Others	30.6
CLK Tree & Pads	**144**
Total	**394**

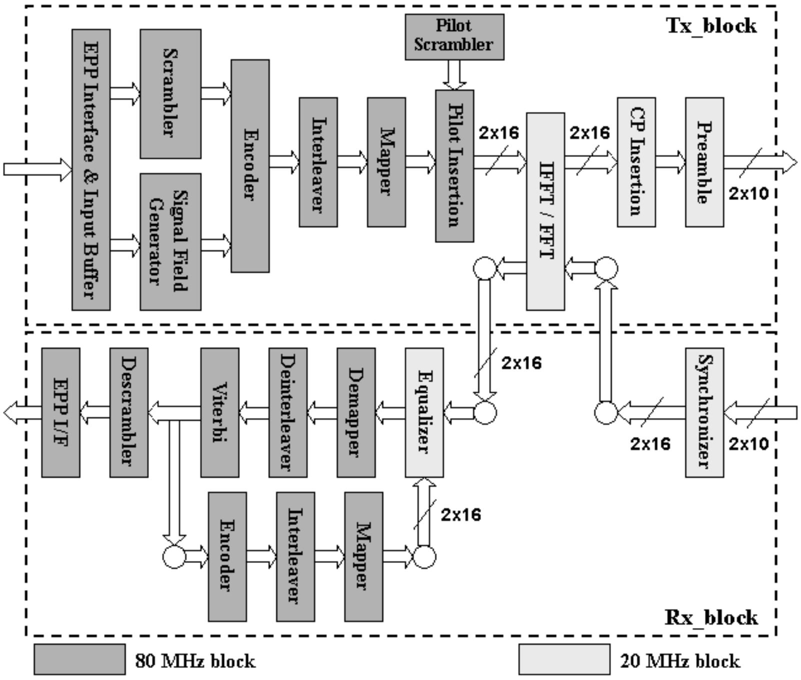


Figure 8.9. Block diagram of the BBP3.

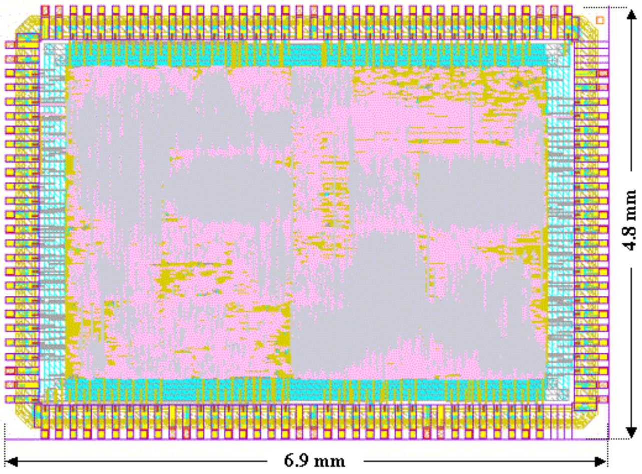


Figure 8.10. Layout of the BBP3.

Table 8.3 shows the power figures obtained for this design from the Synopsys' PrimePower^{TM} tool, which calculates the power consumption based on the switching activity inside the chip under certain operating conditions and is more realistic than the previous obtained power figures, e.g. in Table 8.1. The average powers listed in Table 8.3 were obtained for the 54 Mbps mode. These results are completely in accordance with the power figures reported by leading manufacturers like Atheros [Thom02] or Toshiba [Fuji03]. In the particular case of the Atheros' solution, where a 0.25 μm 5-metal layer technology is used, the core powers in transmit and receive direction are 219 and 203 mW, respectively. Nevertheless, we must say that our BBP3 still does not include two important blocks namely, the DTL for symbol tracking (§7.4) and the NRM in the Channel Estimator (§5.3.2). The latter one is computationally equivalent to an FFT and may increase the power consumption of the Channel Estimator by approximately 50 mW. The BBP3 was taped out in November 2003 and is expected to be ready for testing at the beginning of April 2004.

8.4 Summary

The VLSI realization of the Inner Receiver has been treated in the present Chapter. In order to reduce the power consumption of the Synchronizer, its architecture has been divided into two mutually exclusive data paths, namely the Processing data path and the Tracking data path. The latter is active as long as a frame is being expected. Once the frame is detected and the carrier frequency offset estimation is completed, the Tracking data path gets disabled and the Processing data path operates until the end of the frame. Following this procedure, the estimated power consumption for the Synchronizer using the Synopsys Design Analyzer tool is 140 mW.

The integration of the Inner Receiver was carried out in three steps. The first design, referred to as BBP1, included only the signal processing described in the 802.11a standard plus some blocks of the Channel Estimator. The Synchronizer was not integrated. Due to the amount of buffering used in the design and our lack of experience in the floorplanning process, the resulting IC occupied an area of 107 mm^2 and consumed 1 W of power (estimated).

An improvement of the BBP1 is the BBP2, in which both Synchronizer and Channel Estimator are integrated together. This second experimental chip occupies an area of 59 mm^2 (nearly 50% less than BBP1) with an expected power consumption of 860 mW in the receive direction and 795 mW in the transmit direction. Although the significant reduction with respect to the BBP1, the power figures for the BBP2 are still too large considering that our aim is the full integration of the PHY and DLC layers in a single chip with a targeted power consumption of 1 W altogether (see Chapter 1). The DLC alone is expected to consume about 500 mW [Pan03], and the AFE about 385 mW [Klatt03] (either transmit or receive direction), therefore the power figures given for the BBP2 should be reduced by about 70%.

The BBP3 was developed to achieve this power reduction. In order to simplify the testing of the chip, the EPP interface as well as the Interpolation/Decimation filters were eliminated from the design. The BBP3 results in an area of approximately 34 mm^2, with a power consumption of 104 mW in transmit direction and 146 mW in the receive direction according to the estimations realized by the Synopsys PrimePower^{TM} tool, which are considered to be fairly realistic. In addition, the clock tree and the pads consume altogether 144 mW. These power figures are in accordance with results reported by renowned manufacturers in this field like Atheros or Toshiba.

When integrating analog and digital circuits together on the same substrate, the digital components are seen as the main source of EMI (Electromagnetic Interference) for the AFE. Aiming to reduce the EMI, several investigations are being carried out applying the GALS (Globally Asynchronous Locally Synchronous) concept to the digital design [Krstic03c]. This will be the main feature to be tested in the BBP4 chip.

Chapter 9

Conclusions

THE OFDM TRANSMISSION SCHEME will play a key role in the definition of the PHY layer in future wireless standards due to its natural robustness against multipath propagation. Nevertheless, OFDM demands also computationally complex algorithms, which are necessary to achieve sub-carrier orthogonality during transmission. In this Dissertation, the most common impairments affecting the orthogonality of an OFDM signal have been analyzed in detail. All the impairments can be compensated in one or another way by appropriately designing the AFE or by finding suitable digital estimation and correction algorithms. On this regard, we focused our work on digital techniques in the so-called Inner Receiver. The Inner Receiver is a virtual block included in the receiver, which is responsible to assure that the OFDM symbols obtained after FFT operation are as similar as possible to the original symbols found prior to IFFT calculation on the transmitter side. Therefore, the Inner Receiver will carry out two computationally demanding operations namely, Synchronization and Channel Estimation. Our aim has been to find an efficient (i.e. low-power, low-latency) solution for the Inner Receiver for the special case of the IEEE 802.11a standard. This standard is taken as a reference in this Dissertation, since future definitions of WLAN standards will be very much based on this particular standard.

Several techniques have been analyzed to carry out synchronization. Firstly, the so-called NDA techniques were investigated but were promptly discarded due to the required long estimation time, computational complexity and small estimation range. The so-called DA techniques are more suitable for wireless applications, in which a number of known preamble symbols are appended at the beginning of each transmitted frame. Frame detection, carrier frequency offset estimation, symbol timing estimation and reference CTF estimation are then carried out by means of the preambles. The realization of our low-power low-latency Synchronizer is based on a careful consideration of three design levels:

- *Architectural level*. According to the structure of the preambles, the proposed architecture minimizes latency, since no data buffering is required. All the necessary operations are finalized before reception of data symbols. The Synchronizer is able to estimate a normalized frequency offset in a range of ±1.5 with an error variance of 0.01 using only two autocorrelators.
- *Component level*. Once the structure is defined, the next step consists on the design of each single component in the Synchronizer. There are four key components in this implementation namely, Arctangent operator, NCO, Crosscorrelator and Frame Detector.
  - The Arctangent operator and the NCO are both based on the circular CORDIC algorithm working in vectoring and rotational mode, respectively. In this Dissertation, a new architecture for CORDIC has been investigated, in which no scale factor compensation stage is necessary. The implemented 16-bit rotational CORDIC (acting as a NCO) is able to adaptively select the required rotations and therefore requires up to 50% less iterations on average compared to the classical CORDIC. For both modes of operation the algorithm has a convergence range over the entire coordinate space and has been implemented in form of a pipeline structure in order to reduce latency.
  - The Crosscorrelator is the block used for fine timing estimation and is a weak point in the Synchronizer due to its area and power consumption. In this implementation a variation has been suggested in which only the sign information of the signal is used.
  - Finally, our novel Frame Detector is based on a “plateau detector” that has been designed to be robust against both, noise amplification in low SNR scenarios and AGC over-amplification in high SNR situations.
- *Digital synthesis level*. The last optimization level corresponds to the synthesis of the digital circuit. At this stage, the proposed architecture has been divided into two mutually exclusive datapaths namely, Tracking datapath and Processing datapath. Clock gating has been applied to both datapaths in order to save power. Hence, each datapath is only activated when required and disabled afterwards.

Regarding the Channel Estimator, several options have been investigated. Initially, an experimental system similar to the IEEE 802.11a was used, in which a bigger number of pilot tones was inserted. Results showed that pilot interpolation is not adequate in OFDM systems with a small number of sub-carriers due to the edge effects. Among all the tested interpolation methods, the Low-Rank approach provided the best results.

However, it is not possible to realize pilot interpolation in the particular case of the 802.11a standard due to the small number of pilots per symbol. In this case, Decision-Directed methods similar to the one proposed by Mignone & Morello in [Mig96] are more adequate, since they require no pilots at all, but just initial channel estimation. Two mechanisms are exploited here to reduce the channel estimation error:

1. The error correction capability of the Viterbi decoder. Computer simulations have shown that the TBL in the Viterbi decoder has to be at least 50 bits long.
2. The Low-Rank approach, which exploits the fact that the Channel Impulse Response must be shorter than the Cyclic Prefix. In this case, the “raw” channel estimations are multiplied by a pre-stored Noise Reduction Matrix (NRM) that achieves an improvement of the SNR by about 6 dB.

The small amount of pilots per symbol found in the 802.11a standard can be used to track phase variations derived from synchronization errors. We propose to combine a Decision-Directed (DD) channel estimator with a novel method to estimate the Residual Phase Error (RPE) in which the linear phase error is reduced to a nearly constant phase. The method is seen to perform as well as the direct solution based on Arctangent calculation plus NCO. Nevertheless, the proposed mechanism is an “a posteriori” method in the sense that only the phase errors are corrected but not their origin.

In this sense, computer simulations have shown that sampling clock frequency offsets cannot be left uncorrected. From the proposed RPE estimator an error signal is derived, which is further used as the input to a sampling clock offset estimation and correction algorithm. The sampling clock frequency offset is corrected by digital means through a variable interpolator, which is placed immediately after the ADCs. Due to the nature of the RPE error signal, only first order Farrow interpolators can be used. Higher order filters resulted in instability of the Digital Timing Loop.

Three key components apart from the RPE estimation block can be further identified in our Channel Estimator:

1. FFT/IFFT processor. The proposed solution is based on a conversion of the 64-point FFT into a two-dimensional 8x8-point FFT, in which only 49 non-trivial multiplications are necessary. The parallel-to-parallel operation is carried out in 23 cycles.
2. Complex divider. The proposed pipeline divider is based on the linear CORDIC operated in the vectoring mode. One real division is realized in 16 clock cycles.
3. Viterbi decoder. The proposed soft Viterbi decoder considers four soft-bits and a TBL of 40 bits, which is below the bound of 50 bits obtained through simulations. The reason for that is a technology problem, since no RAM cells were available during synthesis and the traceback memory had to be generated using D flip-flops. This is still an open point in our implementation.

The final result of this Dissertation has been the development of several versions of the Baseband Processing chip namely, BBP1, BBP2 and BBP3, making use of IHP’s 0.25 μm 3/5-metal layer SiGe:C BiCMOS process. Various attempts have been necessary until a Baseband Processing chip with power levels in accordance with our objectives has been realized. Our BBP3 chip consumes 104 mW in the transmitter core, 146 mW in the receiver core and 144 mW in the clock tree and pads. Nevertheless, two important components have not been included in the BBP3 chip namely, the NRM and the DTL. Both may increase the power consumption of the receiver core by about 60 mW and will be integrated in our next BBP4 chip.

The BBP3 is just one of the components in IHP’s targeted single-chip implementation of an 802.11a compliant modem in which the Baseband Processor is integrated together with the RF Front-End and the DLC Processor aiming a maximum power consumption in the order of 1 W. Research is still ongoing in order to achieve the full integration of all the components.

The deployment of OFDM in future communication systems is closely related to its application in the so-called MIMO (Multiple-Input Multiple-Output) systems. In such systems several transmit and receive antennas are used for transmission. Telatar [Tel95] and Forschini [For98] demonstrated theoretically that the MIMO channel capacity increases linearly with the minimum number of transmit or receive antennas.

Nevertheless, this result is only applicable in a Rich-scattering channel, i.e. when the channel coefficients are complex Gaussian random variables, and the signal bandwidth is also small. It is at this point where OFDM finds its purpose, since it is able to convert a wideband channel into N parallel narrowband channels. Hence, future MIMO-OFDM systems will contain N parallel MIMO channels, being each of these channels represented by a *channel matrix*.

Obviously, the power requirements for such systems increase dramatically when compared to the traditional SISO (Single-Input Single-Output) systems. On one hand, synchronization needs to be performed on each signal being received from each antenna. Now the selection of suitable preamble structures is again a key point in order to simplify the implementation of the synchronizer. On the other side, a channel matrix inversion will be necessary during channel equalization. Under this perspective, the results presented in this Dissertation should be taken as a starting point for further investigations regarding the efficient implementation of the Inner Receiver in MIMO-OFDM systems.

Appendix A

Design of a Circular CORDIC Processor

A.1 Introduction

The present Appendix is committed to the realization of two crucial components found in the proposed Synchronizer in Chapter 6, namely the Arctangent calculator and the Numerically Controlled Oscillator. The former block is used to obtain the frequency estimation whereas, the latter is committed to its correction. Both architectures are based on the circular CORDIC processor. The modified CORDIC solutions proposed in the next lines are specially suited for WLAN applications in which latency is a key parameter. The results of this Appendix have been partly reported in [Maha04a] and [Maha04b].

A.2 The CORDIC Algorithm

The COordinate Rotation DIgital Computer (CORDIC) algorithm was firstly developed by J. E. Volder in 1959 [Vold59]. The algorithm is well suited for VLSI realization, being nowadays implemented in common pocket calculators and as arithmetic co-processors in today's general-purpose processors. As its name suggests, the algorithm allows the computation of coordinate rotations in different modes of representation, namely circular, linear and hyperbolic.

Given a vector $v = [x\ y]^T$, the generalized form of plane rotations $R_m(\theta)$ can be expressed by the following linear transformation,

$$R_m(\theta) = \begin{pmatrix} \cos(\theta\sqrt{m}) & \sqrt{m}\sin(\theta\sqrt{m}) \\ \dfrac{-1}{\sqrt{m}}\sin(\theta\sqrt{m}) & \cos(\theta\sqrt{m}) \end{pmatrix}, \tag{A.1}$$

with m = 1 for circular, m = –1 for hyperbolic and m → 0 for linear rotations, respectively (see Figure A.1). In all the cases the transformation $R_m(\theta)[v] = R_m(\theta)\cdot v$ preserves the *norm* of *v*, which is defined as

$$|v| = \sqrt{x^2 + \mathrm{m} \cdot y^2} \, . \tag{A.2}$$

For our purpose, it is of major interest to find an optimized structure allowing the computation of the Arctangent function as well as the complex multiplication of a complex number by a normalized phasor, i.e. a pure vector rotation. As it will be seen, the former operation can be implemented using a circular CORDIC algorithm working in the *vectoring* mode, whereas the latter is implemented through a circular CORDIC algorithm working in the *rotation* mode. The difference between the rotation and vectoring modes of operation will be clarified further on in this Chapter. For the time being, our analysis will be restricted to the circular rotational CORDIC.

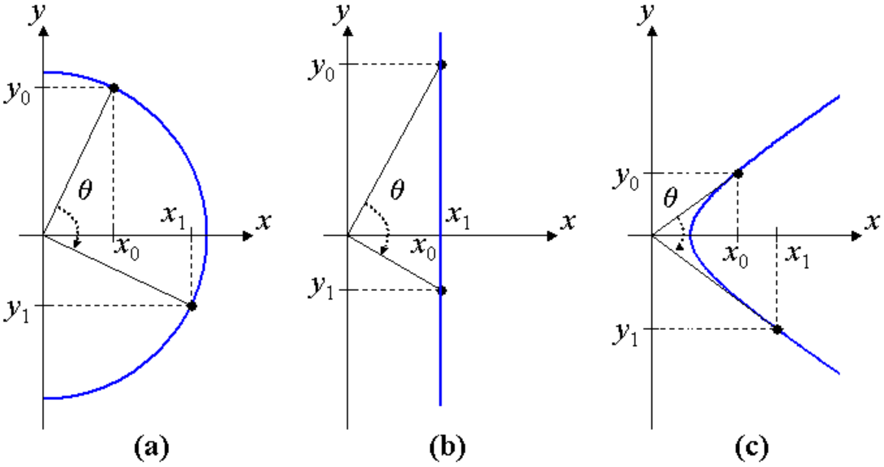


Figure A.1. Plane rotations: a) Circular (m = 1); b) Linear (m → 0); c) Hyperbolic (m = –1).

A.2.1 Circular CORDIC Operation (Euclidean Space)

By letting m = 1 in (A.1), the circular rotation can be represented by the matrix

$$\mathrm{R}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} . \tag{A.3}$$

The matrix R(θ) satisfies two important properties:

1. Being $\theta = \alpha_1 + \alpha_2$, the rotation $\mathrm{R}(\theta) = \mathrm{R}(\alpha_1 + \alpha_2) \equiv \mathrm{R}(\alpha_1)\mathrm{R}(\alpha_2)$, which is expressed as

$$\begin{pmatrix} \cos(\alpha_1+\alpha_2) & \sin(\alpha_1+\alpha_2) \\ -\sin(\alpha_1+\alpha_2) & \cos(\alpha_1+\alpha_2) \end{pmatrix} = \begin{pmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ -\sin(\alpha_1) & \cos(\alpha_1) \end{pmatrix} \begin{pmatrix} \cos(\alpha_2) & \sin(\alpha_2) \\ -\sin(\alpha_2) & \cos(\alpha_2) \end{pmatrix} . \tag{A.4}$$

2. The order of the operation in (A.4) is interchangeable, since the same result is obtained if the vector is firstly rotated by α_1 and afterwards by α_2, or vice versa. This can be expressed as

$$\mathrm{R}(\alpha_1)\mathrm{R}(\alpha_2) = \mathrm{R}(\alpha_2)\mathrm{R}(\alpha_1) \, . \tag{A.5}$$

In the definition of the conventional CORDIC algorithm, the rotation of any vector $v = [x\ y]^{\mathrm{T}}$ through an arbitrary angle θ, $\theta \in [-\pi/2, \pi/2]$, is achieved by expressing θ in terms of few elementary angles satisfying the following condition,

$$\theta = \sum_{i=0}^{b-1} \sigma_i \cdot \alpha_i \,. \tag{A.6}$$

The elementary angles α_i are expressed in general as

$$\alpha_i = \frac{1}{\sqrt{\mathrm{m}}} \tan^{-1}(\delta_i \sqrt{\mathrm{m}}) \,, \tag{A.7}$$

where δ_i is an arbitrary increment or decrement quantity. In the circular mode of operation, m = 1, (A.7) is written as

$$\alpha_i = \tan^{-1}(\delta_i) \,. \tag{A.8}$$

Generally, the δ_i are chosen as inverted powers of 2, i.e. $\delta_i = 2^{-i}$, yielding

$$\alpha_i = \tan^{-1}(2^{-i}) \,. \tag{A.9}$$

Now, remembering the additive property of R(θ) in (A.4), the rotation may be expressed as

$$\begin{aligned}
\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \cos(\sigma_0\alpha_0) & \sin(\sigma_0\alpha_0) \\ -\sin(\sigma_0\alpha_0) & \cos(\sigma_0\alpha_0) \end{pmatrix} \begin{pmatrix} \cos(\sigma_1\alpha_1) & \sin(\sigma_1\alpha_1) \\ -\sin(\sigma_1\alpha_1) & \cos(\sigma_1\alpha_1) \end{pmatrix} \cdots \\
&\quad \cdots \begin{pmatrix} \cos(\sigma_{b-1}\alpha_{b-1}) & \sin(\sigma_{b-1}\alpha_{b-1}) \\ -\sin(\sigma_{b-1}\alpha_{b-1}) & \cos(\sigma_{b-1}\alpha_{b-1}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\
&= \cos(\alpha_0)\cos(\alpha_1)\cdots\cos(\alpha_{b-1}) \begin{pmatrix} 1 & \tan(\sigma_0\alpha_0) \\ -\tan(\sigma_0\alpha_0) & 1 \end{pmatrix} \begin{pmatrix} 1 & \tan(\sigma_1\alpha_1) \\ -\tan(\sigma_1\alpha_1) & 1 \end{pmatrix} \cdots \\
&\quad \cdots \begin{pmatrix} 1 & \tan(\sigma_{b-1}\alpha_{b-1}) \\ -\tan(\sigma_{b-1}\alpha_{b-1}) & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} .
\end{aligned} \tag{A.10}$$

Substituting the value of α_i in (A.9) into (A.10) yields,

$$\begin{aligned}
\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} &= \prod_{i=0}^{b-1} \cos(\alpha_i) \begin{pmatrix} 1 & \sigma_0 2^{0} \\ -\sigma_0 2^{0} & 1 \end{pmatrix} \begin{pmatrix} 1 & \sigma_1 2^{-1} \\ -\sigma_1 2^{-1} & 1 \end{pmatrix} \cdots \\
&\quad \cdots \begin{pmatrix} 1 & \sigma_{b-1} 2^{-(b-1)} \\ -\sigma_{b-1} 2^{-(b-1)} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} ,
\end{aligned} \tag{A.11}$$

where it has been considered that the parameters σ_i in (A.6) are chosen to be either +1 or –1.

In practice, the matrix operations depicted in (A.11) are carried out from left to right, i.e. starting from $i = 0$, and at each step, the value of σ_i corresponding to the elementary angle α_i is determined by the instantaneous error. This error is defined as the difference between the desired angle and the already achieved angle, being

$$\sigma_i = sign\left(\theta - \sum_{r=0}^{i-1} \alpha_r \right) . \tag{A.12}$$

The algorithm described above essentially means that the target angle is achieved adaptively by a *to-and-fro* motion of the vector. Thus, after rotation through one particular elementary angle, the angle to be achieved is calculated by subtracting the already achieved angle from the target angle. If the sign of the residual angle, i.e. the value of σ_i, is –ve, then the vector is rotated through the next elementary angular step in the clockwise direction (–ve direction), otherwise it is rotated in the counter clockwise direction (+ve direction). The operation is repeated until the target angle is reached or until the error between the obtained angle and the target angle is smaller than a predetermined value. The accuracy of the CORDIC is set by the elementary angle corresponding to $i = b - 1$. In Figure A.2, the hardware structure of the basic rotation unit in the conventional circular CORDIC is represented. There, a new variable z has been defined, which operates as an angle accumulator. This variable is initialized with the target angle θ, and it is updated at each iteration step. Therefore, the algorithm will finish its operation when the variable z becomes a machine zero.

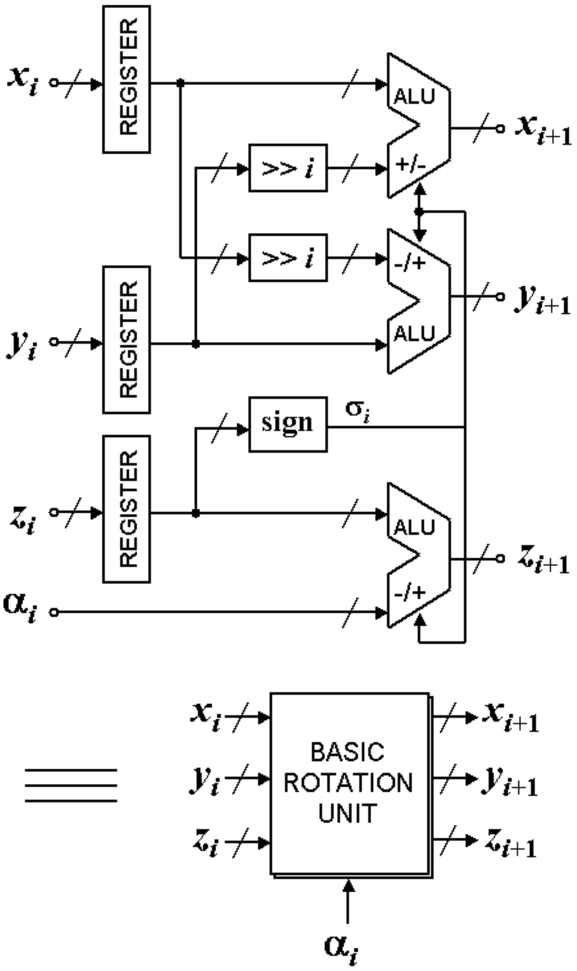


Figure A.2. Block diagram of the basic rotation unit in the conventional circular CORDIC.

For the actual implementation of the CORDIC processor, two possible structures can be considered: a) Folded or recursive structure, or b) Unfolded or pipelined structure [Dawid99], as shown in Figures A.3 and A.4, respectively. In case of the recursive solution, several drawbacks can be mentioned. Firstly, the basic rotation unit shown in Figure A.2 requires variable shifters, whose structure is much more complex compared to fixed shifters. Furthermore, a ROM is necessary in order to store all the possible values for α_i, imposing a limitation in the maximum speed achievable by the algorithm since these memories have long access times. Yet another inconvenient regarding the timing is that for each calculation, the algorithm requires b clock cycles (considering that each iteration step is finished in one clock cycle), but no new calculation can be started until the previous one was finished.

A pipelined structure has its main inconvenient in the required area (*b* basic rotation units are used instead of one), although the total increase in area cannot be said to be *b*-times, since in this case each rotation unit can be very much optimized. Nevertheless, there is a big advantage in a pipelined structure in terms of timing. Once an input vector and the corresponding rotation angle are provided, the pipelined CORDIC requires *b* cycles in order to provide the final solution. Nevertheless, this latency only occurs once, when the operation is started. Afterwards, new sets of data [*x*, *y*, *z*] can be provided at the input and the final results will come *b* cycles later, but one after another at each clock cycle.

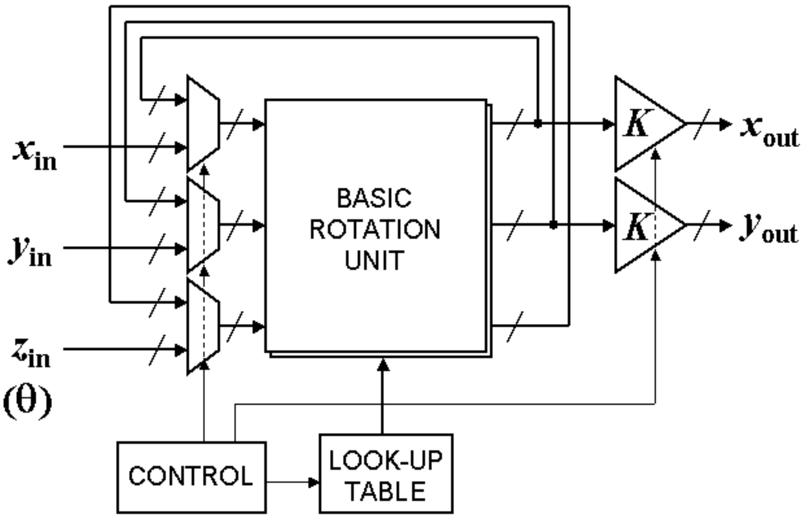


Figure A.3. Circular CORDIC algorithm implemented in form of a recursive structure.

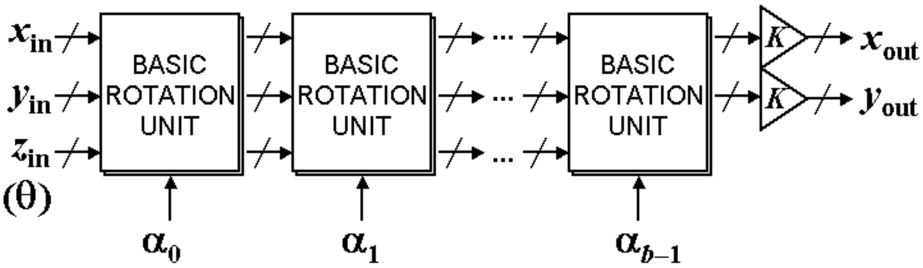


Figure A.4. Circular CORDIC algorithm implemented in form of a pipelined structure.

Whichever solution is taken, the final result has to be multiplied by a scaling factor, which is a machine constant as long as the CORDIC operation undergoes all *b* iterations. The expression for the scaling factor is given in (A.11). This multiplication can be simplified if an approximation of the scaling factor by negative powers of two is done. In this case, the multiplication is realized by shift-and-add operations, but still the number of required cycles for computation is generally high.

Note that, if the algorithm were able to adaptively select the number of iteration steps, a recursive solution would be able to reduce the computation time by avoiding some of the iteration steps. In the same way, a pipelined solution would be able to reduce the power consumption by bridging some of the stages accordingly. The drawback is that in an adaptive CORDIC, the scale factor is no longer a machine constant and its compensation requires an effort comparable to the whole CORDIC algorithm itself.

A.2.2 Modified Circular CORDIC

In order to design our CORDIC processors, we make use of the interesting results obtained by Maharatna in [Maha00]. The modified CORDIC proposed there achieves the target angle by rotating the vector either in clockwise or counter-clockwise direction through small elementary angles, thus avoiding the to-and-fro approach.

In the analysis, only the first terms of the Taylor's series expansion for the sine and cosine functions are taken into account, since these elementary angles are considered sufficiently small. These series are given by the following expressions,

$$\sin(\alpha_i) = \alpha_i - \frac{\alpha_i^3}{3!} + \frac{\alpha_i^5}{5!} - \cdots, \tag{A.13a}$$

$$\cos(\alpha_i) = 1 - \frac{\alpha_i^2}{2!} + \frac{\alpha_i^4}{4!} - \cdots. \tag{A.13b}$$

Using the approximation $\alpha_i = 2^{-i}$, equations (A.13a) and (A.13b) yield

$$\sin(\alpha_i) = 2^{-i} - \frac{2^{-3i}}{6} + \frac{2^{-5i}}{120} - \cdots, \tag{A.14a}$$

$$\cos(\alpha_i) = 1 - \frac{2^{-2i}}{2} + \frac{2^{-4i}}{24} - \cdots. \tag{A.14b}$$

Hence, for small angles the sine and cosine can be taken as

$$\sin(\alpha_i) \approx 2^{-i}, \tag{A.15a}$$

$$\cos(\alpha_i) \approx 1 - \frac{2^{-2i}}{2} = 1 - 2^{-(2i+1)}. \tag{A.15b}$$

It can be readily seen that the largest term being neglected is the second term in the sine series, which is

$$\frac{2^{-3i}}{6} = 2^{-(3i+\log_2 6)} = 2^{-(3i+2.585)}, \tag{A.16}$$

with $0 \le i \le b-1$.

From the above, if $(3i + 2.585)$ equals or exceeds b, the multiplication of any quantity by (A.16) will essentially become machine zero since this operation physically means that the multiplicand gets a right shift equal to or greater than b-bits. Therefore, the condition for preserving the accuracy during such approximation is as follows,

$$3i + 2.585 \ge b, \tag{A.17}$$

which finally results in the following lower bound,

$$i \ge ceiling\left(\frac{b-2.585}{3}\right) = p. \tag{A.18}$$

Nevertheless, the previous bound for i can be slightly relaxed since the alternative terms of equation (A.13a) have a +ve sign. Thus, the greatest term neglected for the above mentioned approximation is not exactly $\alpha_i^3/3!$, but slightly smaller than that. From this point of view, for practical implementation it is sufficient to set

$$i \geq floor\left(\frac{b-2.585}{3}\right) = p\,. \tag{A.19}$$

Considering equations (A.15a) and (A.15b), with the condition in (A.19), the rotation operation in (A.3) can be rewritten as

$$\mathrm{R}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} = \prod_{i=p}^{b-1} \begin{pmatrix} 1-2^{-(2i+1)} & 2^{-i} \\ -2^{-i} & 1-2^{-(2i+1)} \end{pmatrix}. \tag{A.20}$$

Similarly to the result obtained in (A.11), the expression in (A.20) can be realized in practice using only shift-and-add operations. More interestingly, no scaling factor appears in (A.20).

The idea to approach the target angle through a sequence of sufficiently small angular steps shows a serious drawback in the sense that only small target angles can be computed by the modified CORDIC. As an example, consider the case of having 16-bit words. Being b = 16, the value p given in (A.19) is p = 4. The maximum angle that can be computed in this case becomes

$$\theta = \pm\sum_{i=4}^{15} 2^{-i} = \pm 0.1249695\,\mathrm{rad} = \pm 7.1602239^{\circ}\ . \tag{A.21}$$

In order to increase the computation range, a solution is proposed in [Maha00] based on a *domain folding* technique. In this technique, the coordinate space is divided into 16 domains, as shown in Figure A.5.

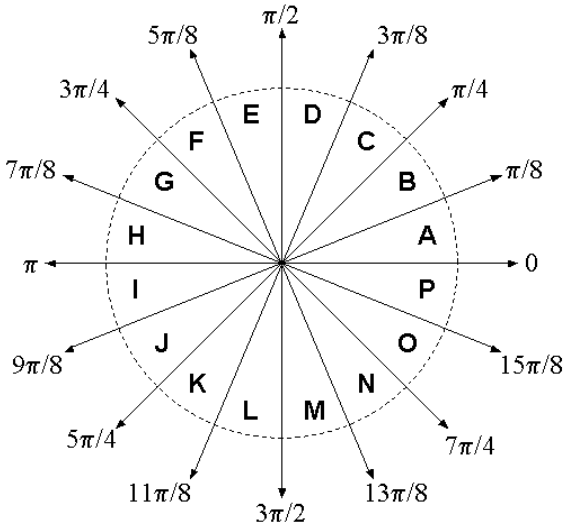


Figure A.5. Partitioning of the coordinate space into 16 domains.

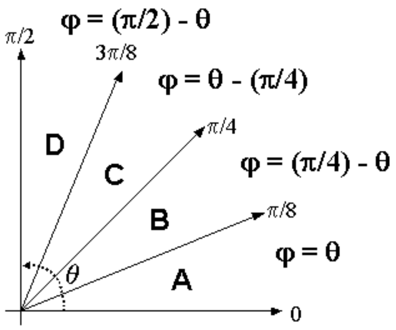


Figure A.6. Partitioning of the first quadrant into four domains.

Initially, a target angle θ restricted to the first quadrant is considered, as shown in Figure A.6. A variable φ can then be defined, which will be restricted into the range [0, π/8]. For each domain A∈[0, π/8), B∈[π/8, π/4), C∈[π/4, 3π/8) and D∈[3π/8, π/2] in the first quadrant, the relationship between θ and φ is given by

$$\theta = \varphi \quad \text{in domain A} \tag{A.22a}$$

$$\theta = \pi/4 - \varphi \quad \text{in domain B} \tag{A.22b}$$

$$\theta = \pi/4 + \varphi \quad \text{in domain C} \tag{A.22c}$$

$$\theta = \pi/2 - \varphi \quad \text{in domain D} \tag{A.22d}$$

Thus, the rotation of any input vector $[x\ y]^T$ by a target angle θ∈[0, π/2], considering this rotation in the clockwise direction (i.e. –ve), can be expressed for each domain as follows,

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{in domain A} \tag{A.23a}$$

$$\begin{pmatrix} x_B \\ y_B \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos(\varphi)+\sin(\varphi) & \cos(\varphi)-\sin(\varphi) \\ -(\cos(\varphi)-\sin(\varphi)) & \cos(\varphi)+\sin(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{in domain B} \tag{A.23b}$$

$$\begin{pmatrix} x_C \\ y_C \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos(\varphi)-\sin(\varphi) & \cos(\varphi)+\sin(\varphi) \\ -(\cos(\varphi)+\sin(\varphi)) & \cos(\varphi)-\sin(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{in domain C} \tag{A.23c}$$

$$\begin{pmatrix} x_D \\ y_D \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \sin(\varphi) & \cos(\varphi) \\ -\cos(\varphi) & \sin(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{in domain D} \tag{A.23d}$$

The above equations can be written in expanded form as

$$\begin{aligned} x_A &= x\cos(\varphi) + y\sin(\varphi) \\ y_A &= -x\sin(\varphi) + y\cos(\varphi) \end{aligned} \tag{A.24a}$$

$$\begin{aligned} x_B &= \frac{1}{\sqrt{2}}\big((x\cos(\varphi) - y\sin(\varphi)) + (x\sin(\varphi) + y\cos(\varphi))\big) \\ y_B &= \frac{1}{\sqrt{2}}\big(-(x\cos(\varphi) - y\sin(\varphi)) + (x\sin(\varphi) + y\cos(\varphi))\big) \end{aligned} \tag{A.24b}$$

$$x_{\mathrm{C}} = \frac{1}{\sqrt{2}}\big((x\cos(\varphi) + y\sin(\varphi)) + (-x\sin(\varphi) + y\cos(\varphi))\big)$$
$$y_{\mathrm{C}} = \frac{1}{\sqrt{2}}\big(-(x\cos(\varphi) + y\sin(\varphi)) + (-x\sin(\varphi) + y\cos(\varphi))\big) \tag{A.24c}$$

$$x_{\mathrm{D}} = x\sin(\varphi) + y\cos(\varphi)$$
$$y_{\mathrm{D}} = -(x\cos(\varphi) - y\sin(\varphi)) \tag{A.24d}$$

Now, remembering that the CORDIC rotation for +φ and –φ is given by the following expressions

$$\begin{pmatrix} x_+ \\ y_+ \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{A.25a}$$

$$\begin{pmatrix} x_- \\ y_- \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{A.25b}$$

respectively, equations (A.24a-d) can be written as

$$x_{\mathrm{A}} = x_- \quad y_{\mathrm{A}} = y_- \qquad (\varphi = \theta) \tag{A.26a}$$

$$x_{\mathrm{B}} = \frac{1}{\sqrt{2}}(x_+ + y_+) \quad y_{\mathrm{B}} = \frac{1}{\sqrt{2}}(-x_+ + y_+) \qquad (\varphi = \pi/4 - \theta) \tag{A.26b}$$

$$x_{\mathrm{C}} = \frac{1}{\sqrt{2}}(x_- + y_-) \quad y_{\mathrm{C}} = \frac{1}{\sqrt{2}}(-x_- + y_-) \qquad (\varphi = \theta - \pi/4) \tag{A.26c}$$

$$x_{\mathrm{D}} = y_+ \quad y_{\mathrm{D}} = -x_+ \qquad (\varphi = \pi/2 - \theta) \tag{A.26d}$$

The above expressions show that the CORDIC operation, with target angles lying in different domains in the first quadrant, can be computed from the results of a CORDIC rotation with target angle $\varphi \in [0, \pi/8]$ by simple addition and subtraction operations, i.e. the domains B, C and D are *folded back* into domain A. A consequence of the domain folding is the appearance of a 1/√2 scaling factor in the case of domains B and C, which can be realized using shift-and-add method within an acceptable error.

The domain folding technique can be extended to the other domains of the coordinate plane in Figure A.5 as shown in Table A.1.

Table A.1. Results of domain folding.

Domain	x_f	y_f	Domain	x_f	y_f	Domain	x_f	y_f	Domain	x_f	y_f
A	x_A	y_A	E	x_A	$-y_A$	I	$-x_A$	$-y_A$	M	$-x_A$	y_A
B	x_B	y_B	F	x_B	$-y_B$	J	$-x_B$	$-y_B$	N	$-x_B$	y_B
C	x_C	y_C	G	x_C	$-y_C$	K	$-x_C$	$-y_C$	O	$-x_C$	y_C
D	x_D	y_D	H	x_D	$-y_D$	L	$-x_D$	$-y_D$	P	$-x_D$	y_D

A.2.3 Pipelined Adaptive Circular CORDIC (Rotation Mode)

Based on the results obtained in §A.2.2, it is possible to extract a pipelined CORDIC structure which is adaptive in nature and whose scaling factor is constant. In our future designs we will consider the case of a 16-bit fixed-point algorithm. As shown previously in §A.2.2, in such a design the index i ranges from 4 to 15. Yet, one problem remains unsolved since the achievable range of the modified CORDIC is ±7.16º (A.21), but the domain folding technique can only restrict the angles into the range ±22.5º. This can be readily tackled by considering a pipeline structure with more than one basic rotation unit corresponding to the index $i = 4$. The number of basic rotation units for $i = 4$ is upper bounded by the maximum possible excursion of the target angle, which is 22.5º (π/8). The difference 22.5º – 7.16º = 15.34º indicates the necessary number of basic rotation units for the index $i = 4$ as

$$L_{(i=4)} = \mathit{ceiling}(15.34^{\circ}\cdot(\pi/180^{\circ})\cdot 2^{4}) + 1, \tag{A.27}$$

resulting in $L_{(i=4)} = 6$.

For the indices $i = 5\ldots15$, only one basic rotation unit is necessary, meaning that the whole pipeline structure requires 17 stages. This seems to be a disadvantage when compared to the traditional CORDIC solution, since the latter requires 16 stages. Nevertheless, as long as the target angle is in the range [0, π/8], the situation where all 17 stages have to be used will never occur. If all stages were activated, it would mean that the angle

$$\theta = 6\cdot 2^{-4} + \sum_{i=5}^{15} 2^{-i} = 25.06^{\circ} > 22.5^{\circ} \tag{A.28}$$

has to be calculated, which obviously falls outside the allowed range.

Another misleading point is the fact that the modified basic rotation units require four add-subtract units instead of the three ones found in the traditional CORDIC (see Figure A.2). This will be true for those values of i such that $2i+1 < b$, i.e. $i = 4\ldots7$ (first 9 stages). For $i \geq 8$, the right-shift operations in the form $(2i+1)$ become a machine zero, since they will be larger than the word length.

These two kinds of modified basic rotation units are shown in Figures A.7 and A.8 referred as Type I and II, respectively. In both cases, through the demultiplex/multiplex stages, it is possible to bridge the input with the output when no rotation is required.

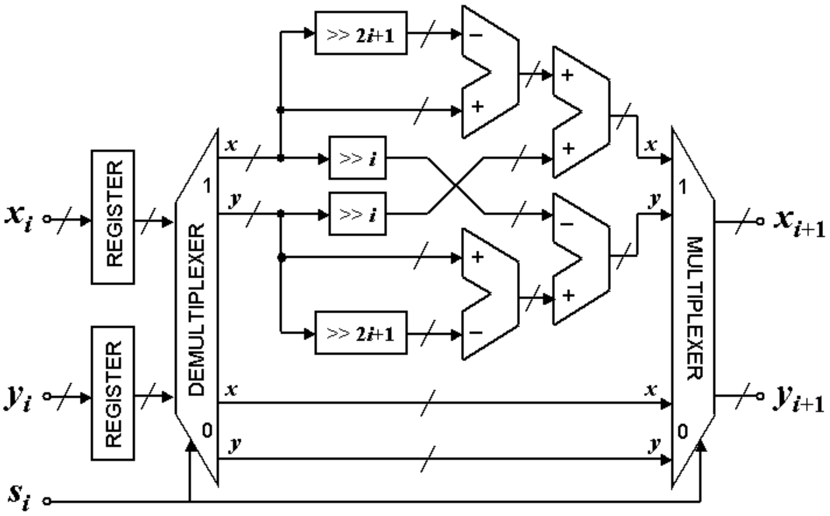


Figure A.7. Block diagram of the modified basic rotation unit (type I, $i = 4\ldots7$).

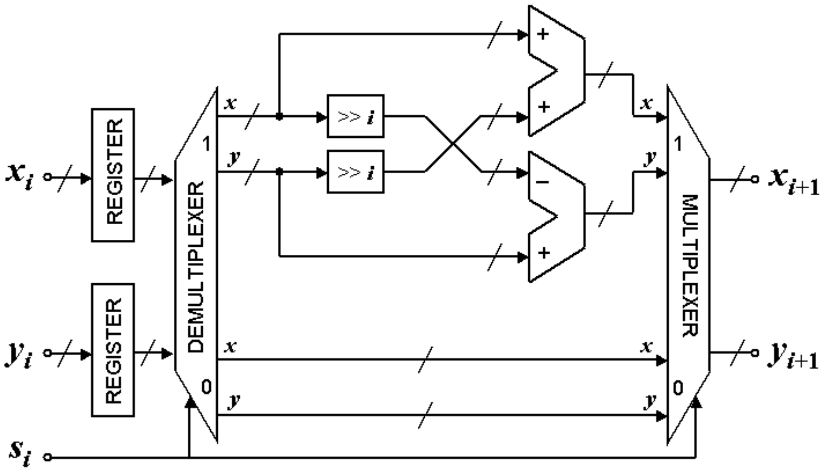


Figure A.8. Block diagram of the modified basic rotation unit (type II, *i* = 8…15).

However, one important problem still remains open namely, how to select the necessary rotation units when a target angle is given. For our representation we selected binary two's complement fixed-point number system. The decimal point was set at the 15th bit position (2^{14}), which means that the interval $[-\pi, +\pi)$ can be represented using 17 bits, as shown in Figure A.9. Considering this bit representation, the value of $+\pi$ is given as 01100100100001111 and $-\pi$ as 10011011011110001. However, the maximum angle allowed at the CORDIC processor is $\pi/8$, whose binary value can be obtained as a right shift by three positions of the binary value of π, i.e. 00001100100100001. Thus, only 13 bits are necessary to represent the target angle (considered positive), i.e. 1100100100001, with the most significant bit representing 2^{-2} and the least significant bit representing 2^{-14}. Using this representation, the three most significant bits, i.e. the ones corresponding to 2^{-2}, 2^{-3} and 2^{-4}, are used to decide the number of active stages with $i = 4$. For the particular case of $\pi/8$, these values correspond to $110_b = 6$, i.e. all six stages with $i = 4$ should be activated.

MSB **LSB**

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure A.9. Selected two's complement fixed-point binary representation of the interval $[-\pi, +\pi)$.

Hence, the decimal value of the three most significant bits directly gives the number of active stages with $i = 4$. A combinatorial decoding block can easily perform this operation. The remaining bits from 2^{-5} to 2^{-14} are directly connected to the input s_i of each elementary rotation unit (see Figures A.7 and A.8). The worst case will happen when the angle $1011111111111 = 5 \cdot 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12} + 2^{-13} + 2^{-14} = 0.3749$ (radian) = 21.48º has to be calculated, which will require 15 active stages for the rotation of the input vector.

The modified CORDIC processor will be mainly composed of three blocks: a Sign/Domain detection circuitry, the pipelined adaptive CORDIC unit, and an Output unit, as shown in Figure A.10. The output unit will apply, when necessary, the scale factor compensation by $1/\sqrt{2}$ as given in (A.26b/c). This compensation is done using shift-and-add operations considering $2^{-1/2} = 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-14}$. The adaptive CORDIC unit is shown in more detail in Figure A.11. There, the Type II rotation units ($i = 7 \ldots 14$) shown in Figure A.8 have been merged into groups of two in order to balance the delay in the pipeline structure.

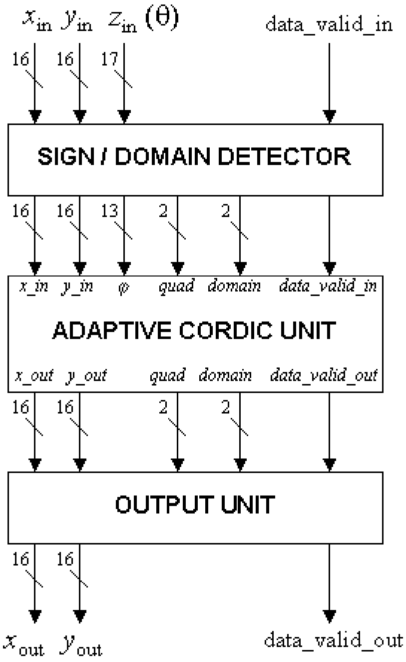


Figure A.10. General block diagram of the adaptive rotation CORDIC processor.

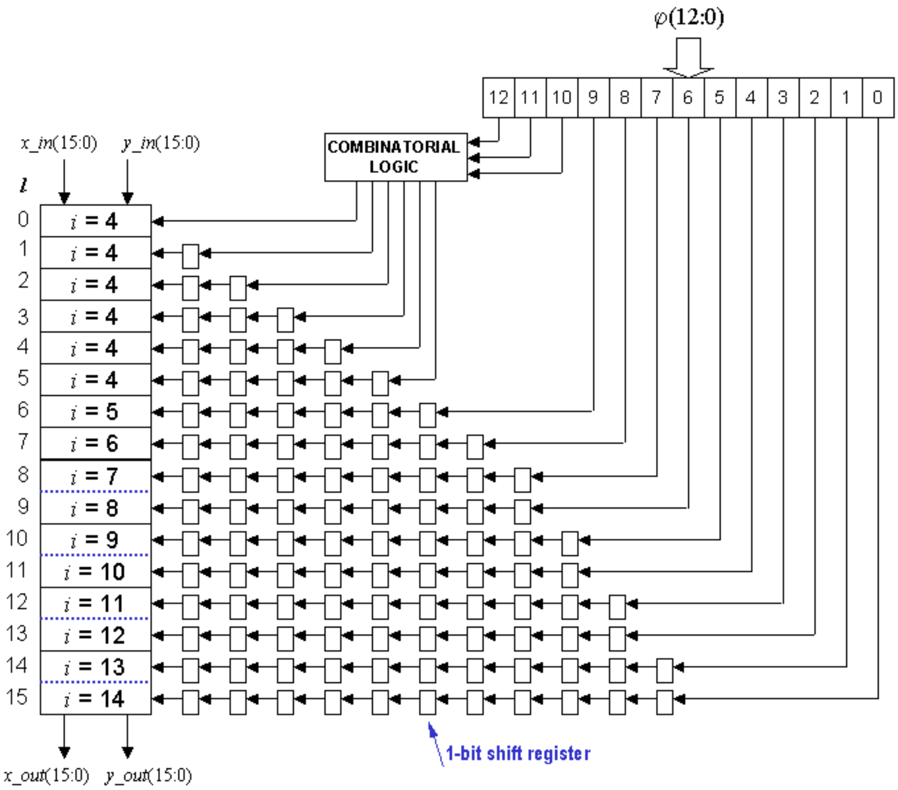


Figure A.11. Block diagram of the pipelined adaptive CORDIC unit.

Figure A.12 shows the number of iterations required by the proposed CORDIC rotator for the angles falling in the range [0, π/8). On average, 8 iterations are necessary, whereas the traditional CORDIC requires 16 iterations in all the cases.

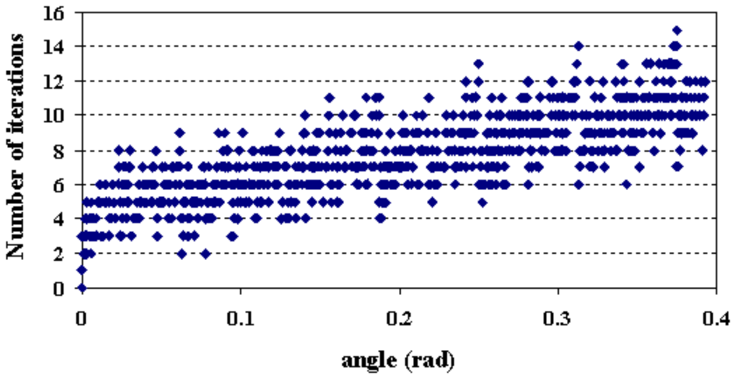


Figure A.12. Number of iterations required by the proposed CORDIC rotator for angles in the range [0, π/8).

A.3 Realization of a CORDIC-based Arctangent Calculator

In the foregoing section the design of the circular CORDIC in the *rotational mode* has been addressed. In this mode of operation, the z input of the CORDIC contains the target angle θ. At each iteration step, the value of z is updated with the already achieved rotation angle, and the whole operation finishes when the accumulated value in z becomes a machine zero. The output vector $[x_{out}, y_{out}]^T$ is therefore the rotated version of the input vector $[x_{in}, y_{in}]^T$ by the angle θ. In the *vectoring mode* of operation, the z input is initialized to 0 and used to accumulate all the rotations necessary until $y_{out} = 0$. Thus, the output x_{out} will finally contain the **magnitude** of the input vector, i.e. $x_{out} = (x_{in}^2 + y_{in}^2)^{1/2}$, whereas z_{out} will contain its **phase**, i.e. $z_{out} = \mathrm{tg}^{-1}(y_{in} / x_{in})$. In Figure A.13 both modes of operation are shown graphically.

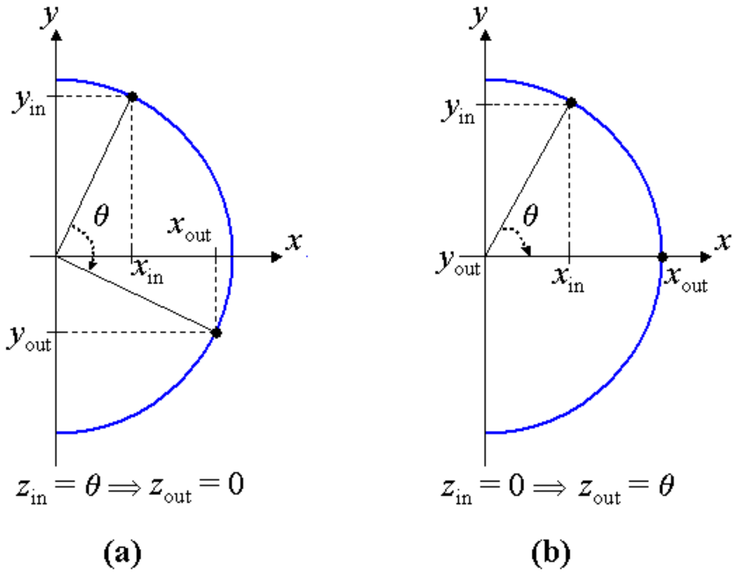


Figure A.13. The circular CORDIC: a) rotational mode; b) vectoring mode.

A.3.1 Domain Folding for Arctangent Calculation

The realization of a CORDIC processor working in the vectoring mode using architectures similar to the ones in Figures A.10-A.11 is possible if the input vector falls in the domain $A \in [0, \pi/8]$ (see Figure A.6). Before the pipelined vectoring CORDIC is presented in §A.3.2, a suitable algorithm for the transformation of the input vectors is derived in this section.

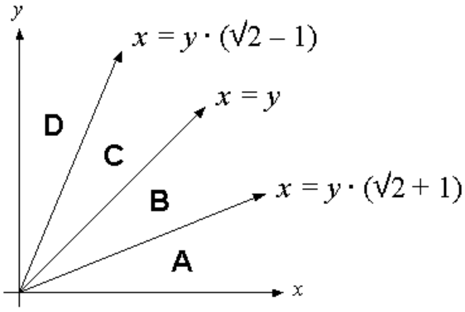


Figure A.14. Relationship between x and y in the domains belonging to the first quadrant.

Let's firstly consider that the input vector lies in the first quadrant, as shown in Figure A.14. There are three borders to be considered, namely $x_{AB} = y \cdot (\sqrt{2} + 1)$, $x_{BC} = y$ and $x_{CD} = y \cdot (\sqrt{2} - 1)$. The original vector has components $[x, y]^T$, whereas the modified vector (CORDIC input) is referred as $[x_{in}, y_{in}]^T$. A possible pseudo-code for the algorithm used to generate $[x_{in}, y_{in}]^T$ based on $[x, y]^T$ would be as follows:

```
if  ($x \geq y$)  then                          -- Domains A or B
        if  ($x \geq x_{AB}$)  then             -- Domain A
                $x_{in} = x$;
                $y_{in} = y$;
        else                                    -- Domain B
                $x_{in} = (x + y)/\sqrt{2}$;    -- (pre-rotation by $\pi/4$)
                $y_{in} = -(y - x)/\sqrt{2}$;   -- The variable $y_{in}$ is forced to be positive
        end if
else                                            -- Domains C or D
        if  ($x < x_{CD}$)  then                -- Domain D
                $x_{in} = y$;                   -- (pre-rotation by $\pi/2$)
                $y_{in} = x$;
        else                                    -- Domain C
                $x_{in} = (x + y)/\sqrt{2}$;    -- (pre-rotation by $\pi/4$)
                $y_{in} = (y - x)/\sqrt{2}$;    -- The variable $y_{in}$ is naturally positive since $y > x$
        end if
end if
```

For the other domains E…P, either x or y or both will be negative. In this case, both variables are forced to be positive, but the sign information is preserved and transferred through the pipeline. The obtained angle after the CORDIC operation should be translated into the correct domain by making use of the domain (A, B, C, D) and sign information. In Table A.2 the corresponding translations are listed, where $\varphi \in [0, \pi/8]$ is the calculated angle and $\theta \in [0, 2\pi)$ the transformed angle. Similarly to what it was done for the rotational CORDIC, the angle φ is represented using 13 bits, which means that the angle θ requires 18 bits.

Table A.2. Final angle transformation after Arctangent calculation.

Domain	θ	Domain	θ	Domain	θ	Domain	θ
A	φ	E	$(\pi/2)+\varphi$	I	$\pi+\varphi$	M	$(3\pi/2)+\varphi$
B	$(\pi/4)-\varphi$	F	$(3\pi/4)-\varphi$	J	$(5\pi/4)-\varphi$	N	$(7\pi/4)-\varphi$
C	$(\pi/4)+\varphi$	G	$(3\pi/4)+\varphi$	K	$(5\pi/4)+\varphi$	O	$(7\pi/4)+\varphi$
D	$(\pi/2)-\varphi$	H	$\pi-\varphi$	L	$(3\pi/2)-\varphi$	P	$2\pi-\varphi$

Considering that the input variables x and y are represented using 16 bits, the values for x_{AB} and x_{CD} can be obtained applying shift-and-add operations on the variable y, taking the following approximations

$$(2^{1/2} - 1) = 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7} + 2^{-13} \tag{A.29a}$$

$$(2^{1/2} + 1) = 2^{1} + 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7} + 2^{-13} \tag{A.29b}$$

A.3.2 Pipelined Circular CORDIC (Vectoring Mode)

As mentioned in §A.3.1, a pipelined structure for the CORDIC algorithm working in the vectoring mode can be derived from the solution found for the rotation mode. However, the solution will not be adaptive since the value of the y variable cannot be made independently zero with respect to φ.

At each stage, the input vector $[x_i, y_i]^T$ has to be rotated according to the value i. After rotation, the sign of the y component has to be checked. If this component is positive, the rotation is considered to be valid and the obtained rotated vector is connected at the output $[x_{i+1}, y_{i+1}]^T$; otherwise ($y \leq 0$), an excessive rotation was applied to the input vector, and therefore the rotation is discarded by bridging the input with the output through a multiplexer. The determination of the sign of y is straightforward since a two's complement representation is used. Furthermore, the sign of y, after being inverted, provides a partial result of the final angle φ. Figures A.15 and A.16 show two possible schemes for the rotation stages. Similarly to the results obtained when designing the rotational CORDIC, the *vectoring units* can be simplified when $2i + 1 \geq b$ (Type II structure). The general scheme of the vectoring CORDIC is shown in Figure A.17. Note that the final angle θ is represented using 18 bits, since the angle is considered to be in the interval $[0, 2\pi)$. Furthermore, Figure A.18 shows the resulting pipeline scheme for the proposed modified circular CORDIC algorithm in the vectoring mode. The design has been realized considering 16-bit input x and y signals and a representation of the angles with the decimal point at 2^{14}. The combinatorial logic found in the pipeline structure is basically an encoding block, which transforms the number of active inputs into its binary equivalent value.

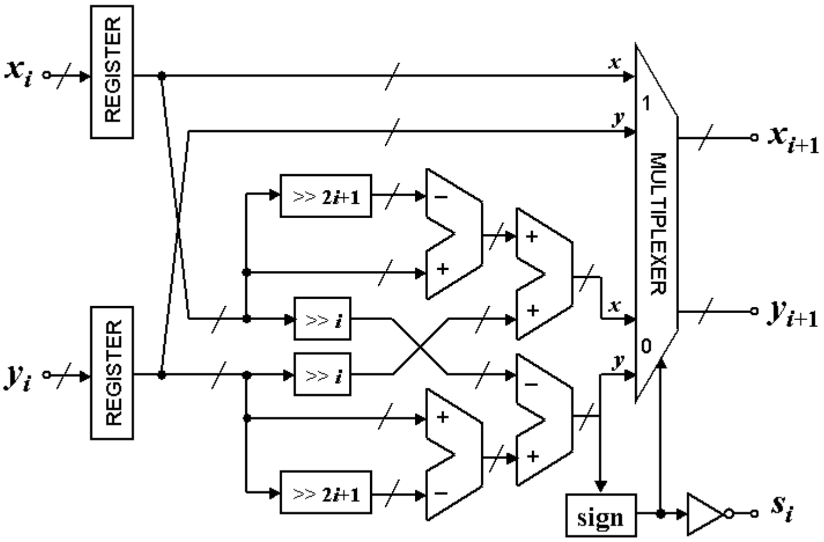


Figure A.15. Block diagram of the modified basic vectoring unit (type I, *i* = 4…7).

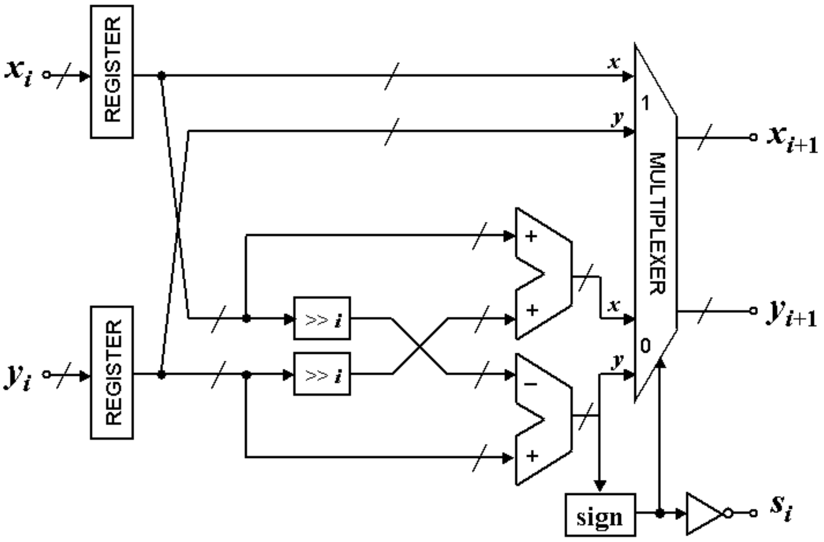


Figure A.16. Block diagram of the modified basic vectoring unit (type II, *i* = 8…15).

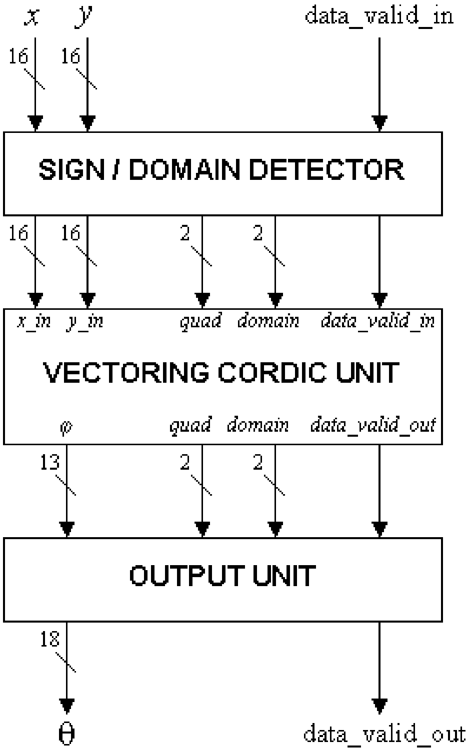


Figure A.17. General block diagram of the Arctangent calculator.

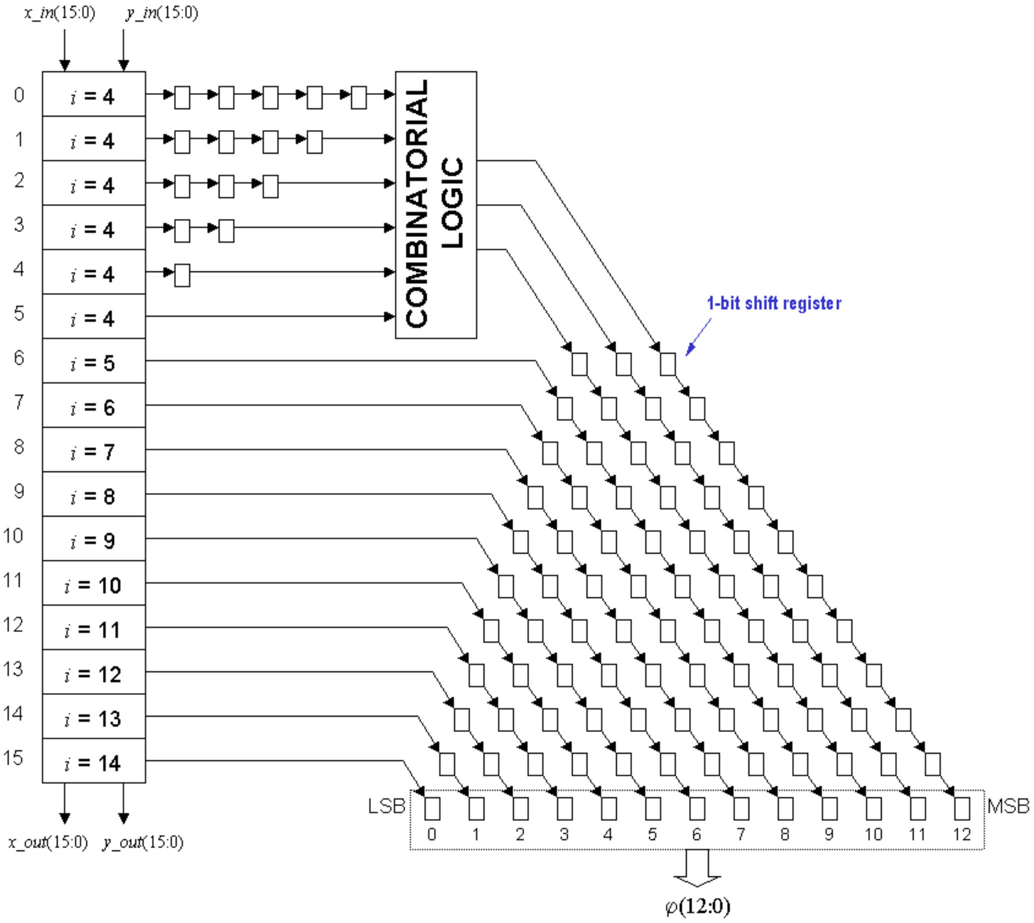


Figure A.18. Detailed scheme of the pipelined Arctangent calculator.

A.4 Realization of a CORDIC-based Numerically Controlled Oscillator

The realization of a Numerically Controlled Oscillator (NCO) is straightforward once the principles of the circular CORDIC (rotation mode) have been presented. For our purpose, the NCO is used in the Synchronizer to eliminate the carrier frequency offset. As it is shown in Chapter 6, this frequency offset is estimated through an Arctangent calculation. If the normalized frequency offset is ε (normalized with respect to the sub-carrier spacing), the result of the Arctangent calculation will be $-2\pi\varepsilon$. The input samples have to be multiplied by the phasor $e^{-j\frac{2\pi}{\mathrm{N}}\cdot\varepsilon\cdot n}$, which represents a pure rotation of the samples by a time-dependent angle (the magnitude is kept constant). The parameter N is the number of sub-carriers used in the OFDM system, and n stands for the discrete time variable, expressed as a modulo N value. As N is generally a power of 2, a division by N represents a right-shift operation.

The value $-2\pi\varepsilon\cdot n/\mathrm{N}$ may correspond to either a –ve rotation (positive ε) or a +ve rotation (negative ε). At each clock cycle, an amount equal to $-2\pi\varepsilon/\mathrm{N}$ is added to the stored rotation angle. A small combinatorial logic checks whether the new rotation angle to be stored goes beyond the interval $[-\pi, \pi]$. If that happens, the angle is modified accordingly. The proposed scheme for the NCO is depicted in Figure A.19. The rotational CORDIC included there corresponds to our modified version shown in Figures A.10 and A.11.

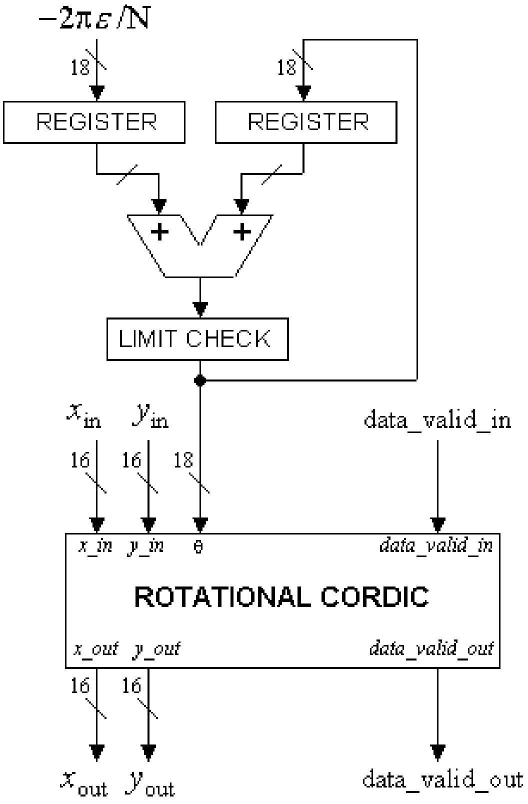


Figure A.19. General scheme for the NCO based on the modified rotational CORDIC.

A.5 Summary

The present Appendix has dealt with the design of a circular CORDIC processor working in both, rotational and vectoring mode of operation. After presentation of the classical approach, a simplification of the problem has been introduced, in which the *scaling factor* no longer appears and a convergence range over the entire coordinate space is achieved through a *domain folding* technique. The proposed CORDIC adaptively converges to the target angle by executing only the needed iteration steps while skipping the others without compromising the accuracy. The design has been customized in a 16-bit CORDIC processor in which iteration savings of up to 50% are achieved on an average. The cell area of the 16-bit rotational CORDIC is only 0.7 mm^2 with an estimated power consumption of 7 mW @ 20 MHz, 2.5 V, using IHP’s 0.25 μm 5-metal layer BiCMOS technology. The latency of the processor is 14 clock cycles and the throughput is one set of result per clock cycle. The design is used in the Synchronizer described in Chapter 6 as a Numerically Controlled Oscillator.

The proposed concept is also applied in the design of a CORDIC processor working in the vectoring mode for Arctangent calculation. However, in this case no adaptive selection of rotations can be applied. The proposed algorithm does also not need any scale factor compensation step. The synthesized cell area is 0.5 mm^2 with an estimated power consumption of 6 mW.

The main drawback of the proposed scheme is that if accuracy is to be increased, larger word lengths and more pipeline stages will be necessary, thus increasing the hardware cost. Moreover, in our theory the primary assumption about the angle depends on the wordlength ($\sin(\alpha_i) = \alpha_i = 2^{-i}$). If we increase the wordlength then the largest angular step that we can allow will be very small –see expression for *p* in (A.19)- and it may result in a very long pipeline. The proposed implementation is not easy to adapt for more than 20 bit internal representation. Instead, some hybrid scheme falling in between the conventional CORDIC and our scheme could be a solution for larger bit representation.

Appendix B

Design of Linear-Phase IIR Interpolation and Decimation Filters

B.1 Introduction

ONE IMPORTANT DESIGN ISSUE WHEN DEALING WITH OFDM SIGNALS is the Inter-Symbol Interference introduced by the several filters employed during A/D and D/A conversion as well as in the RF up- and down-conversion. As explained in Chapter 2/Appendix C, the addition of a cyclic prefix helps to cope with this problem. However, in a real scenario all this filtering will be undistinguishable from the filtering due to the multipath channel. The consequence of this is an effective reduction of the longest path allowed in the multipath channel or, equivalently, a reduction in the allowed size of the room where the transmission takes place. Thus, in order to take the maximum advantage of the cyclic prefix, all filters should be designed to show the shortest possible impulse response.

The condition imposed above is specially challenging for those filters requiring a very sharp transition band, since these filters require larger orders, i.e. longer impulse responses. This is the case for the digital Interpolation and Decimation filters found before the D/A and after the A/D conversion. Hence, FIR structures should be discarded due to the large orders they require. Digital IIR filters may yield the solution, but still they show two main drawbacks:

a) By definition IIR filters have an infinite impulse response. We must restrict ourselves to very low-order filters in which this response can be truncated. This will also reduce problems with stability.

b) IIR filters do not show in general a linear phase in the frequency response inside the pass-band. This is important in order to reduce the EVM.

In this Chapter, a general solution for the design of linear-phase IIR filters based on an algorithm presented by Johansson in [Johan96] will be treated in §B.2. This algorithm will be used in §B.3 to derive the design of our interpolation and decimation filters in the particular case of the 802.11a and HL2 standards. The interpolation/decimation factor selected in our implementation is $N_{INT/DEC} = 4$ ($f_S = 80$ MHz). This high oversampling rate has two main advantages. On one side, the design of the analog AAFs (see Figure 3.1) is simplified, since these filters are allowed to increase their transition bands. This results in lower order filters, which at the same time have a better phase response in the passband. On the other hand, the higher is $N_{INT/DEC}$, the simpler is the design of the low-pass half-band filters found at each stage of interpolation/decimation. In the particular case of the decimation filters, these filters should be placed just after the variable interpolator investigated in Chapter 7.

B.2 Introduction

B.2.1 Main Features of an All-pass Filter

As its name indicates, an all-pass filter is a kind of filter, whose frequency response shows a unit magnitude throughout the entire frequency range $\omega \in [0, 2\pi)$. In order to obtain such a response, it is necessary for each pole in the transfer function to come together with a zero, whose value is the conjugated inverse of the pole. Thus, the general transfer function of an *M*-order all-pass filter is given by

$$H_A(z) = \prod_{i=1}^{M} \frac{z^{-1} - d_i^*}{1 - d_i z^{-1}}. \tag{B.1}$$

Only *stable* all-pass filters are considered, i.e. those filters whose poles fall inside the unit circle (see Figure B.1). An example of such a filter is the pure delay line $H(z) = z^{-K}$ ($K>0$), which has K poles at $z = 0$ and K zeros at $z = \infty$.

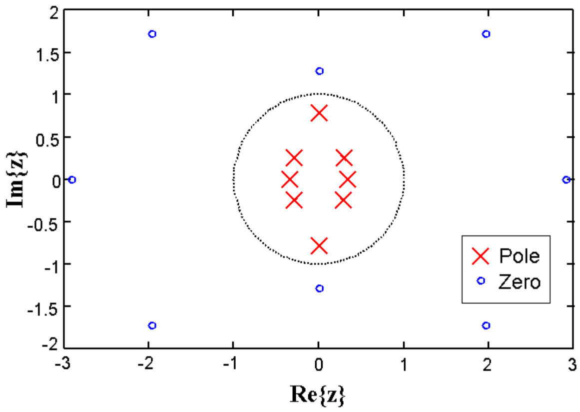


Figure B.1. Pole-zero diagram of a generic all-pass filter with order *M*=5.

In [Mari95] it is shown that the *group delay* of an all-pass filter is positive for all ω. The group delay is defined as follows,

$$\tau(\omega) = -\frac{d}{d\omega} \arg\{H(e^{j\omega})\} \tag{B.2}$$

and is expressed in samples. From (B.2), it can be concluded that for an all-pass filter, the derivative of the argument of $H(e^{j\omega})$ is negative for all ω, meaning that the phase response is monotone decreasing for all ω.

B.2.2 General Filter Representation

For the design of optimized interpolation and decimation low-pass filters, two important results concerning the representation of digital filters are considered here. As explained in [Vaid93], any filter H(z) can be decomposed using a *polyphase representation* as

$$\mathrm{H(z)} = \frac{1}{N}\sum_{i=0}^{N-1} \mathrm{z}^{-i}\mathrm{H}_i(\mathrm{z}^N)\,. \tag{B.3a}$$

The second important result, also derived in [Vaid93], is that a wide family of practical transfer functions including Butterworth, Chebyshev and Elliptic filters can be represented as

$$\mathrm{H(z)} = \frac{\mathrm{H}_{A0}(\mathrm{z}) + \mathrm{H}_{A1}(\mathrm{z})}{2}\,, \tag{B.3b}$$

where $\mathrm{H}_{A0}(\mathrm{z})$ and $\mathrm{H}_{A1}(\mathrm{z})$ are stable unit-magnitude all-pass filters. The expression in (B.3b) is referred to as the *All-pass Decomposition Theorem*.

In order to understand the basic idea behind equation (B.3b), we recall that the all-pass functions have frequency responses given by

$$\mathrm{H}_{A0}(e^{j\omega}) = e^{j\Phi_0(\omega)}\,, \tag{B.4a}$$

$$\mathrm{H}_{A1}(e^{j\omega}) = e^{j\Phi_1(\omega)}\,. \tag{B.4b}$$

Therefore, the filter in (B.3b) can be further expressed as

$$\begin{aligned}\mathrm{H}(e^{j\omega}) &= \frac{e^{j\Phi_0(\omega)} + e^{j\Phi_1(\omega)}}{2} \\ &= e^{j\frac{\Phi_0(\omega)+\Phi_1(\omega)}{2}}\cos\left(\frac{\Phi_0(\omega)-\Phi_1(\omega)}{2}\right).\end{aligned} \tag{B.5}$$

In (B.5) it can be seen that the magnitude of the filter in (B.3b) depends on the phase difference between the all-pass filters, $\Delta\Phi(\omega) = \Phi_0(\omega) - \Phi_1(\omega)$. Therefore, the passband of H(ω) will occur when $\Phi_0(\omega) \approx \Phi_1(\omega)$ whereas, the stopband will occur when $\Phi_0(\omega) - \Phi_1(\omega) \approx \pi$. Both phases, $\Phi_0(\omega)$ and $\Phi_1(\omega)$, are monotone decreasing functions.

B.2.3 Design Algorithm

Trying to combine the properties expressed by equations (B.3a/b), our aim is the design of a low-pass IIR filter with the following polyphase representation ($N = 2$), as given in (B.3a)

$$\mathrm{H_{LP}(z)} = \frac{1}{2}\left(\mathrm{H}_0(\mathrm{z}^2) + \mathrm{z}^{-1}\mathrm{H}_1(\mathrm{z}^2)\right) \tag{B.6}$$

and with the property that the filters $\mathrm{H}_0(\mathrm{z}^2)$ and $\mathrm{z}^{-1}\mathrm{H}_1(\mathrm{z}^2)$ can be selected to be all-pass filters, as given in (B.3b).

The obtained low-pass filter should have an approximately linear phase. This can be achieved by setting an odd number of delays on branch number one [Johan96]. The overall transfer function of this all-pass branch is consequently

$$\mathrm{z}^{-1}\mathrm{H}_1(\mathrm{z}^2) = \mathrm{z}^{-2R+1}\,, \tag{B.7}$$

with $R = 1, 2, \ldots$

The filter orders between the two all-pass branches must differ by one in order to obtain a low-pass characteristic. In addition, the attenuation in the stopband is maximized if the branch with pure delay elements is chosen to have the lowest overall order of the two branches. Therefore, the order of branch zero is chosen to be $2R$. The transfer function of this branch can be written in the form,

$$H_0(z^2) = \frac{\sum_{i=0}^{R} a_i z^{-2(R-i)}}{\sum_{i=0}^{R} a_i z^{-2i}}, \tag{B.8}$$

which is an alternative way to represent an all-pass filter derived from the expression given in (B.1). With these selections, the order of the overall filter $H_{LP}(z)$ in (B.6) becomes $M = 4R - 1$.

Under these circumstances, in order to design a low-pass filter with approximately linear phase in the passband, the following optimization problem has to be solved:

1) $\Phi_0(\omega) = -(2R-1)\omega$; for $\omega \in [0, \omega_P]$
2) $\Phi_0(\omega) = -(2R-1)\omega + \pi$; for $\omega \in [\omega_S, \pi]$

with ω_P and ω_S being the passband and stopband limits, respectively (see Figure B.2).

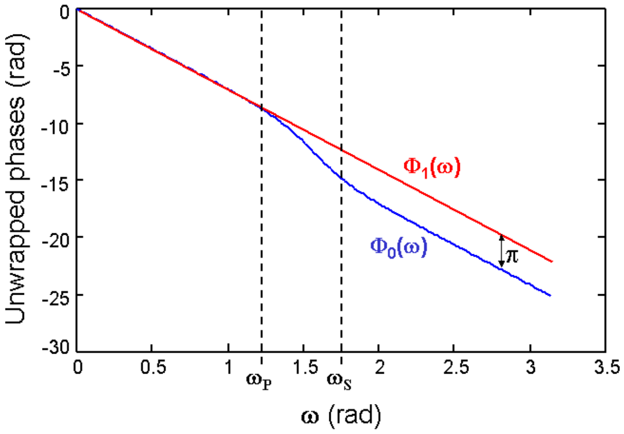


Figure B.2. Example of unwrapped phases for the all-pass filters used to design a low-pass filter.

From the knowledge of $\Phi_0(\omega)$, the coefficients a_i in (B.8) have to be derived. This is a non-trivial optimization problem, which has been tackled by several authors in the literature [Johan96], [Johan97], [Kruk99], [Kruk01], [Law97]. In our case, the solution proposed in [Johan96] has been applied, since it seems to derive the simplest algorithm among all the referenced.

The number of coefficients, and consequently, the number of degrees of freedom for an M^{th} order linear phase low-pass filter is $R=(M+1)/4$. To obtain an attenuation zero at the angle ω_0 in the passband, the phase responses of the two branches must be equal at that angle. Each attenuation zero thus binds one degree of freedom. Hence, for a specific filter order M, which also specifies the number of coefficients R, the filters are designed to have R attenuation zeros inside the passband. Furthermore, a low-pass filter designed in this manner will show an equiripple behavior in the stopband [Johan96].

The algorithm proposed in [Johan96] is now explained in detail. The R locations of the attenuation zeros are used as unknowns in the algorithm. Let the parameter vector Ω denote the set of the angles at which these attenuation zeros occur, i.e.

$$\Omega = [\omega_1, \omega_2, \ldots, \omega_R]^{\mathrm{T}} \tag{B.9}$$

and let $|\mathrm{H}(\Omega, e^{j\omega})|$ denote the magnitude of the low-pass filter as a function of Ω. The algorithm adjusts the locations of the attenuation zeros iteratively until an equiripple behavior in the stopband is achieved. The maximum value of the attenuation in the stopband achievable by the algorithm, i.e. δ_{S}, will depend on the selected order for the filter and is a priori unknown. The algorithm is performed in the following steps:

1. Set the iteration counter $i = 0$. Guess the initial values for the angles in the parameter vector $\Omega^{(0)} = [\omega_1^{(0)}, \omega_2^{(0)}, \ldots, \omega_R^{(0)}]^{\mathrm{T}}$. A good initial guess proposed in [Johan96] is

   $$\omega_r^{(0)} = \frac{1}{2}\sin^{-1}\left(\sin(\omega_{\mathrm{P}})\cdot\sin\left(\frac{\pi\cdot r}{2R+1}\right)\right), \tag{B.10}$$

   with $r = 1, \ldots, R$.
2. Determine the filter coefficients of branch zero such that the phase responses of both branches at the angles in $\Omega^{(i)}$ are equal, i.e.

   $$\Phi_0^{(i)}(\omega_r^{(i)}) = \Phi_1^{(i)}(\omega_r^{(i)}) = -(2R-1)\omega_r;\ \ r = 1, 2, \ldots, R \tag{B.11}$$

   This problem is not obvious and a suitable algorithm will be presented in §B.2.4. Once the filter coefficients have been obtained, calculate the magnitude function $|\mathrm{H}(\Omega^{(i)}, e^{j\omega})|$.
3. Find the $R + 1$ largest maxima of the function $|\mathrm{H}(\Omega^{(i)}, e^{j\omega})|$ in the stopband. Let $\omega_l^{(i)}$, $l = 0, 1,\ldots, R$, denote the $R + 1$ corresponding angles at which these maxima occur.
4. Calculate the quotient Q between the largest and the smallest value of the $R + 1$ maxima of $|\mathrm{H}(\Omega^{(i)}, e^{j\omega})|$ calculated in step 3. That is

   $$\mathrm{Q} = \frac{\max\limits_{0\le l\le R}\{\mathrm{H}(\Omega^{(i)}, e^{j\omega_l^{(i)}})\}}{\min\limits_{0\le l\le R}\{\mathrm{H}(\Omega^{(i)}, e^{j\omega_l^{(i)}})\}}. \tag{B.12}$$

   If $\mathrm{Q} < 1 + \varepsilon$, then stop. Otherwise go to the next step. The value ε can be any small value, which must be specified beforehand. It must be sufficiently small to ensure an equiripple solution.
5. Solve the system of equations

   $$\left|\mathrm{H}(\Omega^{(i+1)}, e^{j\omega_l^{(i)}})\right| = \delta_{\mathrm{S}}^{(i+1)}, \tag{B.13}$$

   with $l = 0, 1, \ldots, R$ and $\Omega^{(i+1)}$ being the parameter vector for the next iteration. Set the iteration counter $i=i+1$, and go to step 2.

   The system of equations to be solved here is a non-linear system. Nevertheless, it is possible to linearize it by considering the following approximation

   $$\left|\mathrm{H}(\Omega^{(i)}, e^{j\omega_l^{(i)}})\right| + \sum_{r=1}^{R}\frac{\partial}{\partial\omega_r^{(i)}}\left|\mathrm{H}(\Omega^{(i)}, e^{j\omega_l^{(i)}})\right|\cdot\Delta\omega_r^{(i)} = \delta_{\mathrm{S}}^{(i+1)}, \tag{B.14}$$

   for $l = 0, 1, \ldots, R$.

The partial derivatives in (B.14) can be computed numerically by changing the values in $\Omega^{(i)}$ slightly and recomputing the values of H. The computation of the r^{th} partial derivative can be performed in the following steps:

1) Set $\Omega' = \Omega^{(i)}$. Replace $\omega_r^{(i)}$ in Ω' by $\omega' = \omega_r^{(i)} + \Delta\omega$.
2) Calculate $|\mathrm{H}(\Omega', e^{j\omega})|$. This can be done in the same way as in step 2 of the design algorithm.
3) Calculate the partial derivative at the angles $\omega_l^{(i)}$, $l = 0, 1, \ldots, R$ as

$$\frac{1}{\Delta\omega}\left(\left|\mathrm{H}(\Omega', e^{j\omega_l^{(i)}})\right| - \left|\mathrm{H}(\Omega^{(i)}, e^{j\omega_l^{(i)}})\right|\right) \tag{B.15}$$

and call the corresponding calculated values h'_{0r}, h'_{1r}, …, h'_{Rr}.

Equation (B.14) can now be written as

$$h_l + \sum_{r=1}^{R} h'_{lr} \cdot \Delta\omega_r^{(i)} = \delta_{\mathrm{S}}^{(i+1)} \tag{B.16}$$

for $l = 0, 1, \ldots, R$, and

$$h_l = \left|\mathrm{H}(\Omega^{(i)}, e^{j\omega_l^{(i)}})\right|. \tag{B.17}$$

This can be written in the following matrix form,

$$\begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_R \end{pmatrix} + \begin{pmatrix} h'_{01} & h'_{02} & \cdots & h'_{0R} \\ h'_{11} & h'_{12} & \cdots & h'_{1R} \\ \vdots & \vdots & \ddots & \vdots \\ h'_{R1} & h'_{R2} & \cdots & h'_{RR} \end{pmatrix} \begin{pmatrix} \Delta\omega_1^{(i)} \\ \Delta\omega_2^{(i)} \\ \vdots \\ \Delta\omega_R^{(i)} \end{pmatrix} = \begin{pmatrix} \delta_{\mathrm{S}}^{(i+1)} \\ \delta_{\mathrm{S}}^{(i+1)} \\ \vdots \\ \delta_{\mathrm{S}}^{(i+1)} \end{pmatrix}. \tag{B.18}$$

The previous system of equations will be solved as

$$\begin{pmatrix} \Delta\omega_1^{(i)} \\ \Delta\omega_2^{(i)} \\ \vdots \\ \Delta\omega_R^{(i)} \end{pmatrix} = \overline{\mathrm{H}}_{\Delta} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_R \end{pmatrix}, \tag{B.19}$$

where

$$\overline{\mathrm{H}}_{\Delta} = \frac{\mathrm{H}_{\Delta}}{\max\{\mathrm{H}_{\Delta}\}} \tag{B.20}$$

and

$$\mathrm{H}_{\Delta} = -\begin{pmatrix} h'_{01} & h'_{02} & \cdots & h'_{0R} & -1 \\ h'_{11} & h'_{12} & \cdots & h'_{1R} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h'_{R1} & h'_{R2} & \cdots & h'_{RR} & -1 \end{pmatrix}^{-1}. \tag{B.21}$$

The normalization of the matrix H_{Δ} in (B.20) is necessary in order to assure the convergence of the algorithm. This normalization is not mentioned in [Johan96], but it is a result of our experiments. The new parameter vector for the next iteration $i + 1$ is therefore

$$\Omega^{(i+1)} = \Omega^{(i)} + \Delta\Omega^{(i)}, \tag{B.22}$$

with

$$\Delta\Omega^{(i)} = \left(\Delta\omega_1^{(i)} \quad \Delta\omega_2^{(i)} \quad \cdots \quad \Delta\omega_R^{(i)}\right)^{\mathrm{T}}. \tag{B.23}$$

B.2.4 Calculation of Filter Coefficients

In step 2 of the algorithm, the filter coefficients are calculated such that (B.11) is satisfied. This can be done in the Laplace domain using recurrence formulas for polynomials [Johan96].

The overall transfer function of the delay branch is z^{-2R+1}. Hence, the phase response values of this branch at the attenuation zeros are

$$\Theta_r = \Phi_1(\omega_r) = -(2R-1)\omega_r \;\; ; \;\; r = 1, 2, \ldots, R. \tag{B.24}$$

The problem now is to determine the coefficients of the all-pass branch filter $H_0(z^2)$ of order $2R$ such that the phase response values at the attenuation zeros equal the phase response values of the delay branch, i.e.

$$\arg\{\mathrm{H}_0(e^{j2\omega_r})\} = \Theta_r \;\; ; \;\; r = 1,2,\ldots,R\,. \tag{B.25}$$

In the Laplace domain (Ψ), the corresponding all-pass function is of the form

$$\mathrm{H}_0(\Psi) = \frac{\mathrm{P}_R(\Psi)}{\mathrm{P}_R(-\Psi)}\,. \tag{B.26}$$

The numerator and denominator polynomials contribute with the same amount to the phase response. It is therefore sufficient to find a numerator polynomial $\mathrm{P}_R(\Psi)$ having half the desired phase response values at the attenuation zeros. Thus, using the bilinear transform, the equations that must be satisfied are

$$\arg\{\mathrm{P}_R(j\Omega_r)\} = \frac{\Theta_r}{2} \;\; ; \;\; r = 1,2,\ldots,R\,, \tag{B.27}$$

where

$$\Omega_r = \tan(\omega_r)\,. \tag{B.28}$$

The polynomial $\mathrm{P}_R(\Psi)$ is generated using recurrence formulas. First, the parameters β_1, β_2, …, β_R are computed using the following recursive continued fraction formula (see [Johan96] and references therein):

$$\begin{aligned}
&\beta_1 = \frac{\tan\left(\frac{\Theta_1}{2}\right)}{\Omega_1}\,; \\
&\beta_2 \cdot \left(\Omega_2^2 - \Omega_1^2\right) = 1 - \frac{\beta_1\Omega_2}{\tan\left(\frac{\Theta_1}{2}\right)}\,; \\
&\beta_r \cdot \left(\Omega_r^2 - \Omega_{r-1}^2\right) = 1 - \cfrac{\beta_{r-1}\cdot(\Omega_r^2 - \Omega_{r-2}^2)}{1 - \cfrac{\beta_{r-2}\cdot(\Omega_r^2 - \Omega_{r-3}^2)}{\vdots \atop 1 - \cfrac{\beta_2\cdot(\Omega_r^2 - \Omega_1^2)}{1 - \cfrac{\beta_1\Omega_r}{\tan\left(\frac{\Theta_r}{2}\right)}}}}\;,\;\; r > 2
\end{aligned} \tag{B.29}$$

The polynomial $P_R(\Psi)$ can then be derived using the following recurrence formula,

$$\begin{aligned} &P_0(\Psi) = 1 \\ &P_1(\Psi) = 1 + \beta_1 \Psi \\ &P_r(\Psi) = P_{r-1}(\Psi) + \beta_r \cdot \left(\Psi^2 + \Omega_{r-1}^2\right) \cdot P_{r-2}(\Psi) \ , \ r = 1,2,\ldots,R \end{aligned} \tag{B.30}$$

The all-pass function $H_0(z^2)$ is considered to be composed of cascaded second and fourth order sections. The transfer function for the second order sections is given by

$$\frac{1-\alpha_1 z^2}{-\alpha_1 + z^2} \tag{B.31}$$

and for the fourth order section by

$$\frac{1-\alpha_3(1-\alpha_2)z^2 - \alpha_2 z^4}{-\alpha_2 - \alpha_3(1-\alpha_2)z^2 + z^4} . \tag{B.32}$$

To calculate the filter coefficients, the numerator polynomial $P_R(\Psi)$ is first factorized into first and second order factors of the form

$$-\Psi + A_{0i} \tag{B.33}$$

and

$$\Psi^2 - A_{1j}\Psi + A_{0j} , \tag{B.34}$$

where the indices indicate the first order factor *i* and the second order factor *j*, respectively. The coefficients for the second and fourth order sections can now be calculated as

$$\alpha_{1i} = \frac{1-A_{0i}}{1+A_{0i}} \tag{B.35}$$

and

$$\alpha_{2j} = \frac{A_{1j} - A_{0j} - 1}{A_{1j} + A_{0j} + 1} , \tag{B.36}$$

$$\alpha_{3j} = \frac{1-A_{0j}}{1+A_{0j}} . \tag{B.37}$$

B.3 Application in Interpolation and Decimation Filters

The design of the A/D and D/A interfaces is shown in Figure B.3. An interpolation/decimation factor $N_{INT/DEC}$ = 4 was selected for implementation. As shown in Figure B.3, the interpolator is divided into two stages with N=2 each. The corresponding low-pass filter $H_{INT}(z)$ is also divided into two stages, where each stage is designed considering a polyphase representation as in (B.3a). Here the noble identities for multirate systems as provided in [Vaid93] are used.

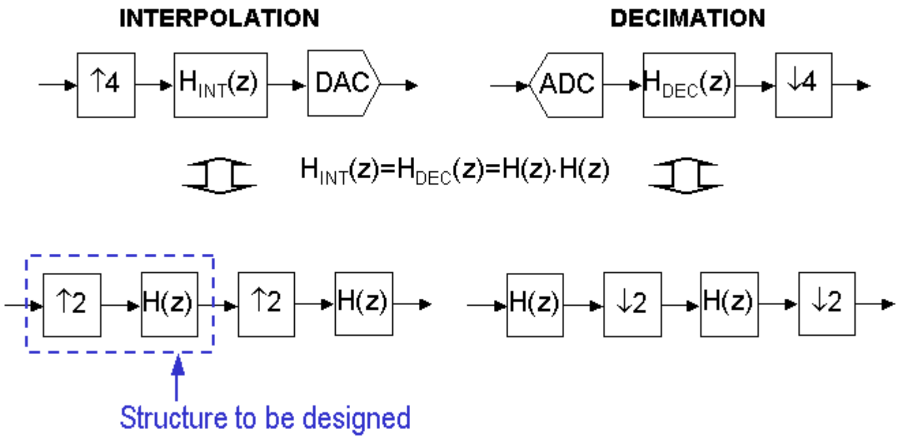


Figure B.3. General scheme of the Interpolation and Decimation problem for N=4.

The identity #2 [Vaid93] is applied in order to establish an alternative way to represent a filter H(z) following an expander block, considering the polyphase representation of that filter. This alternative way is shown in Figure B.4. The advantage of this new scheme is that the filters no longer operate at $2 \cdot f_S$ but at f_S (reference sampling frequency), thus reducing the computation requirements. Only a digital switch placed at the output of the filter has to be operated at $2 \cdot f_S$.

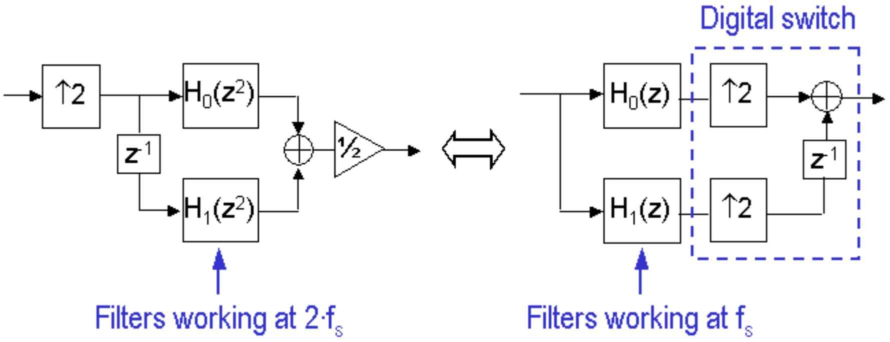


Figure B.4. Simplification through polyphase representation of the basic Interpolation stage.

A similar solution can be found for the decimation filter, now invoking identity #1 [Vaid93]. In this case the digital switch is placed at the input and is used to distribute the input samples alternatively to each of the filters. Moreover, an adding block is necessary at the output. The scheme for the basic decimation stage is shown in Figure B.5.

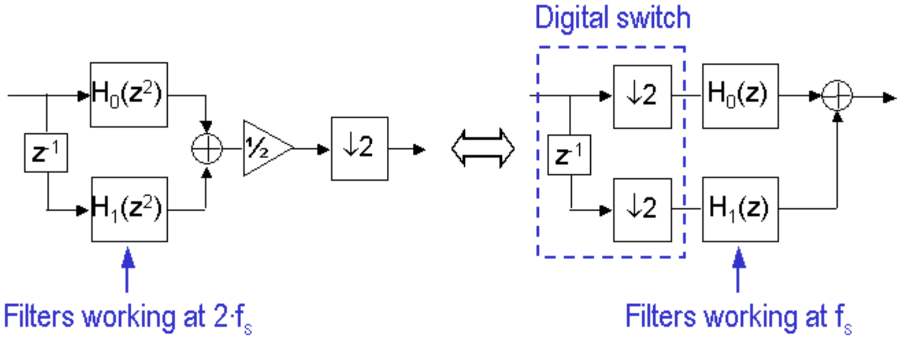


Figure B.5. Simplification through polyphase representation of the basic Decimation stage.

Taking these simplifications into consideration, it is now possible to design the filter H(z) as given in (B.6). According to the algorithm, the lower branch filter $H_1(z^2)$ will be z^{-2R}. The upper branch filter $H_0(z^2)$ has to be determined applying the algorithm. For that, only small values of *R* will be considered, since our interest is not only to keep a nearly constant group delay in the passband, but also to make this group delay as small as possible.

The filter H(z) should be ideally a half-band filter, i.e. the passband should fall in the interval $\omega \in [0, \pi/2]$ and the stopband in the interval $\omega \in (\pi/2, \pi]$, without any transition band. The value of $\omega=\pi/2$ corresponds to an analog frequency of 10 MHz. Fortunately, the standards 802.11a and Hiperlan2 consider only sub-carriers -26 to -1 and +1 to +26 to carry information. Sub-carriers from -32 to -27 and from +27 to +31 are permanently set to 0, thus the bandwidth of the signal is effectively reduced to approximately 8.125 MHz. The input signal to the decimator will be in addition affected by a frequency offset, whose value is supposed to be in the range ±468.75 kHz (see Chapter 6). Thus the bandwidth to be considered should be at least 9 MHz. In our design, a transition band from 9 to 11 MHz has been taken into account. In the half-band filter, this is equivalent to set the cut-off pulsation of the passband at $\omega_P = 2\pi(9/40)$ and the stopband starting at pulsation $\omega_S = 2\pi(11/40)$.

Four possible values for *R* have been inspected in the algorithm, i.e. *R* = 2, 3, 4, 5, which was stopped when the parameter Q in (B.12) was below $1+10^{-5}$. In Figures B.6-B.9 the resulting frequency response of the obtained filter H(z) for the different values of *R* is represented. As expected, this response shows *R* maxima in the stopband as well as *R*+1 attenuation zeros in the passband.

In the case of *R* = 2, an attenuation of 20 dB is achieved in the stopband. This value is small enough considering that two of such filters will be used in practice. Furthermore, the maximum value of attenuation is increased by approximately 5 dB if *R* is increased by 1. This is not much if we consider that an increase of one unit in *R* means an increase of four units in the resulting filter order. Nevertheless, this result is derived from the tight transition band selected for this particular design. For looser transition bands, the attenuation in the stopband can be significantly improved by increasing *R*.

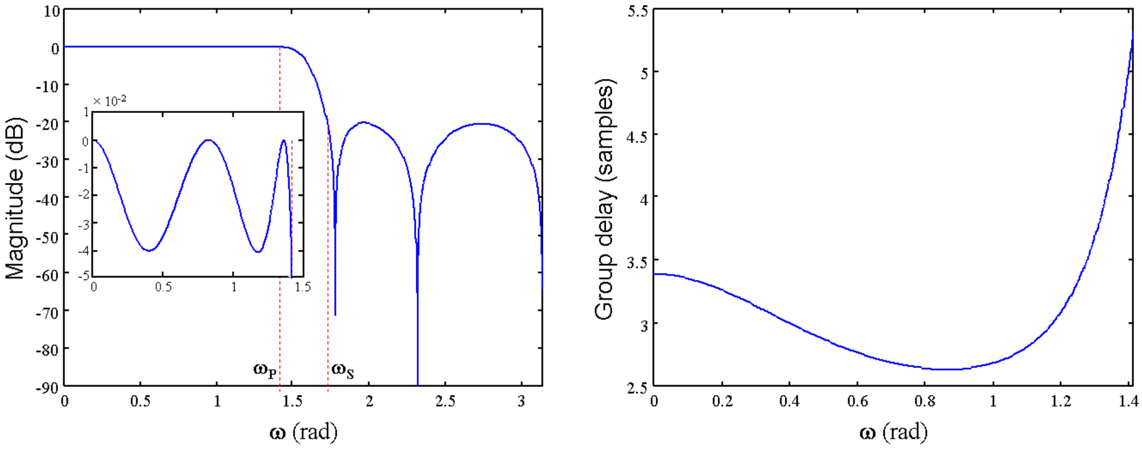


Figure B.6. Resulting low-pass filter H(z) (magnitude and group delay) for *R* = 2.

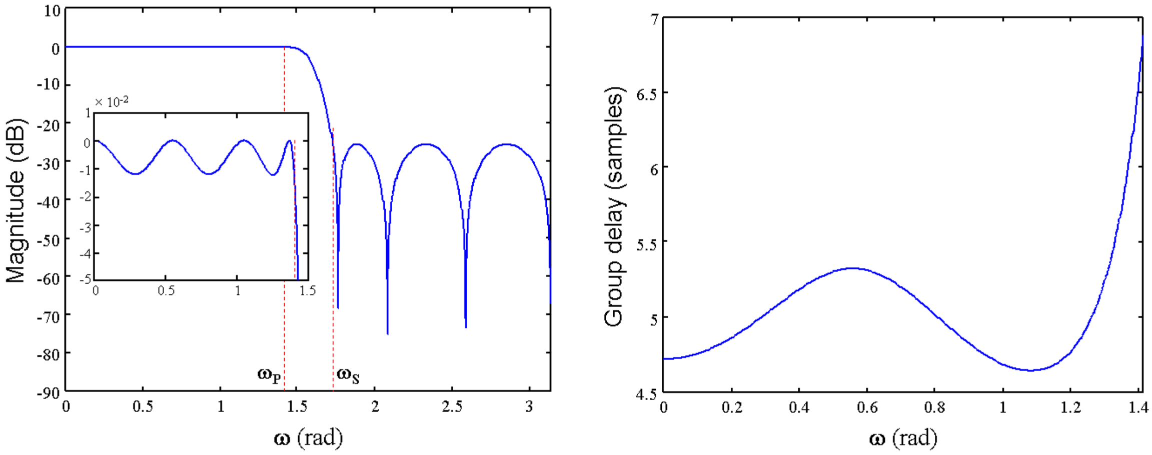


Figure B.7. Resulting low-pass filter H(z) (magnitude and group delay) for R = 3.

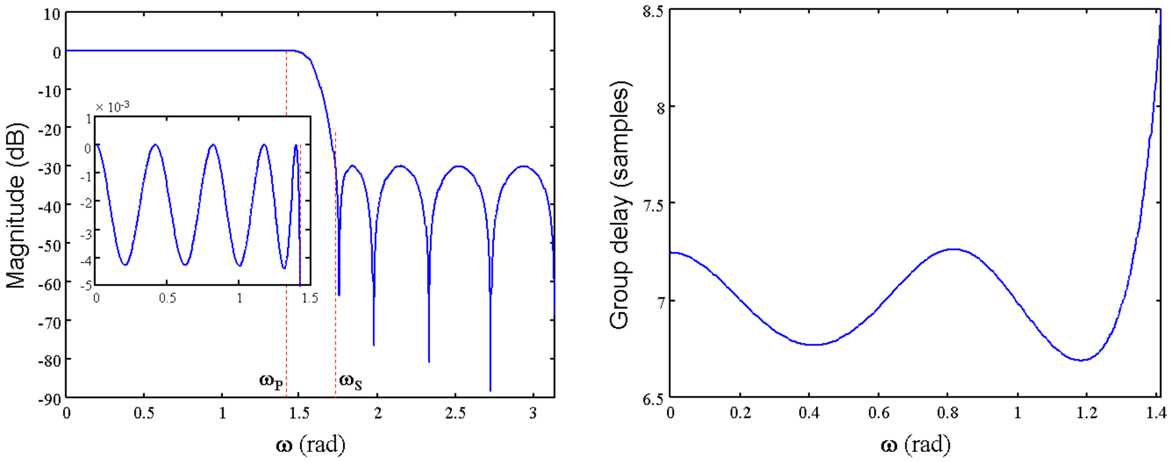


Figure B.8. Resulting low-pass filter H(z) (magnitude and group delay) for R = 4.

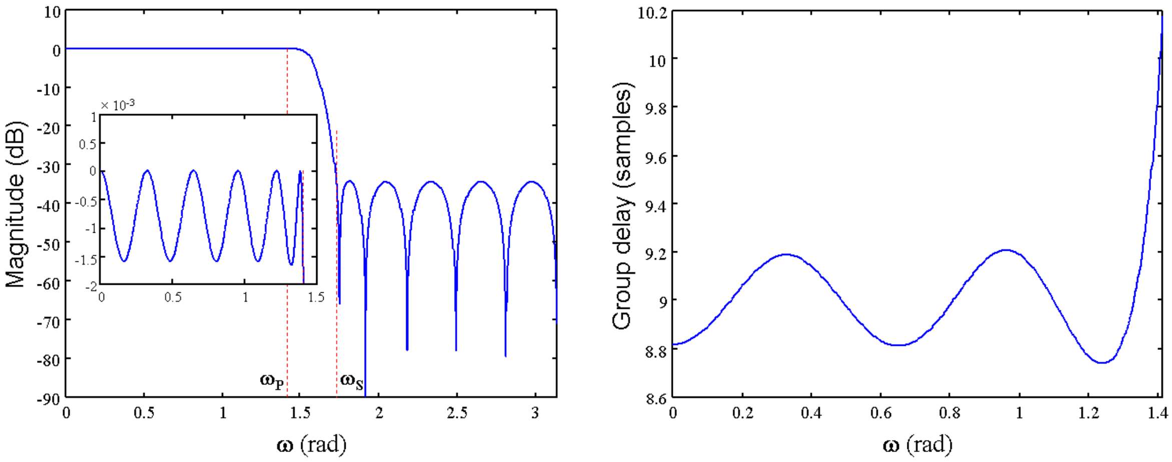


Figure B.9. Resulting low-pass filter H(z) (magnitude and group delay) for R = 5.

The solution with *R* = 2 was finally selected for implementation. In this case, the filters $H_0(z^2)$ and $H_1(z^2)$ in (B.6) resulted as follows,

$$H_0(z^2) = \frac{-0.2023 + 0.4779 \cdot z^{-2} + z^{-4}}{1 + 0.4779 \cdot z^{-2} - 0.2023 \cdot z^{-4}}, \tag{B.38}$$

$$H_1(z^2) = z^{-2}. \tag{B.39}$$

The final design for the Interpolation and Decimation filters is as shown in Figures B.10 and B.11, respectively. These filters correspond to the ones depicted in Figure 8.6 and included in the design of the BBP2. The ADCs and DACs are considered to be 10-bit. However, internally the operations are carried out using 12-bit representation.

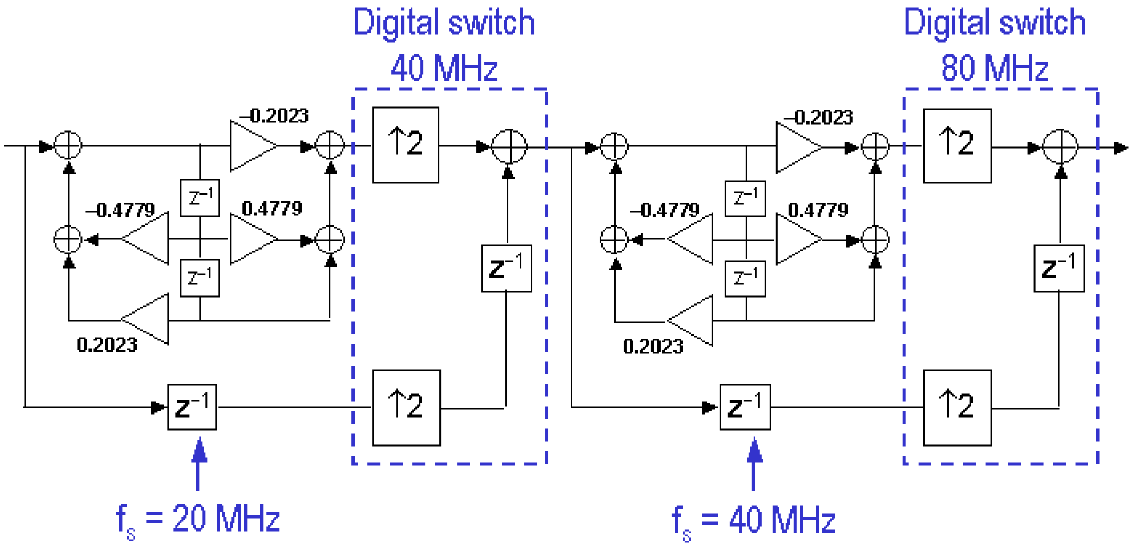


Figure B.10. Block diagram of the proposed Interpolation filter (N_{INT} = 4) included in the BBP2 chip.

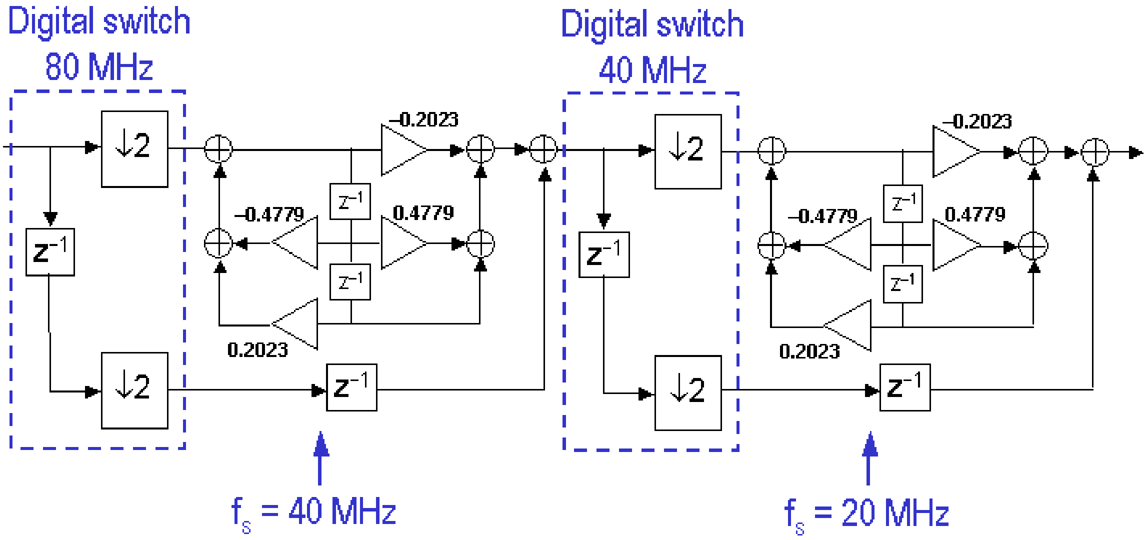


Figure B.11. Block diagram of the proposed Decimation filter (N_{DEC} = 4) included in the BBP2 chip.

B.4 Summary

Phase linearity is a desired feature specially in the case of OFDM signals since a better EVM performance can be achieved. During Interpolation and Decimation with the oversampling factor being a power of two, half-band low-pass filters with sharp transition bands are needed. IIR filters are the best candidates, since they require less filter orders than FIR filters, but generally they do not show a linear phase response in the passband. Therefore, a suitable algorithm to design linear-phase IIR filters obtained as the addition of two all-pass filters has been investigated in the present Appendix. The aim is to design the Interpolation/Decimation filters to be used in our 802.11a Baseband processors.

In our implementation, an oversampling factor $N_{INT/DEC} = 4$ was selected for two main reasons. On one hand, it allows a simpler design of the Interpolation/Decimation filters. Since these operations are performed at two stages, the filters at each stage do not need a big attenuation in the stopband, thus simplifying the design of the filter. In our design case, each filter has an attenuation of about 20 dB in the stopband and the overall attenuation is more than 40 dB. On the other side, this scheme is suitable to be used in a low-IF approach for the AFE, as shown in Figure 3.10. In this way, the problem of the I/Q mismatches is solved and a significant amount of power can be saved in the analog side.

After synthesis, the Interpolator resulted in a cell area of 0.31 mm^2 with an estimated power consumption of 5.6 mW. For the Decimator, the cell area is 0.32 mm^2 and consumes 4.3 mW. These power figures were obtained from the *Synopsys Design Analyzer* tool and generally are pessimistic values. Note that the Decimator should be placed immediately after the variable interpolator used to correct the sampling clock frequency offset.

Indeed, an alternative solution may be envisioned in the case of OFDM signals. Since an IFFT/FFT operation is required, the OFDM symbols may be optimally interpolated and decimated by means of the Fourier Transform, as described in [Vaid93]. In such an approach, a 256-point IFFT/FFT processor working at 80 MHz is required. The way to carry out the interpolation is by appending zero samples to the original 64 sample sequence forming the OFDM symbol. Nevertheless, the obvious drawback in the ideal solution is that not only the IFFT/FFT processor has to be run at 80 MHz but also the Synchronizer itself.

Appendix C

Minimum Cyclic Prefix Extension and Orthogonality

The objective of this Appendix is to demonstrate that with a sufficiently large cyclic prefix extension, OFDM signals can eliminate Inter-Symbol Interference and keep orthogonality. Let's consider the sequence $\mathbf{A_R}=[A_{R0}, A_{R1},\ldots,A_{R\mathrm{N}-1}]^{\mathrm{T}}$ at the receiver side. This sequence can be expressed using matrix notation as follows,

$$\mathbf{A_R} = \mathbf{F^H} \cdot \mathbf{E} \cdot \mathbf{H} \cdot \mathbf{G} \cdot \mathbf{F} \cdot \mathbf{A}, \tag{C.1}$$

where $\mathbf{A}=[A_0,\ldots,A_{\mathrm{N}-1}]^{\mathrm{T}}$ is the transmitted sequence, $\mathbf{F^H}$ is the N-by-N point DFT matrix, $\mathbf{F}$ is the N-by-N point IDFT matrix, $\mathbf{H}$ is the channel matrix with dimensions (N+G+L–1)-by-(N+G), and **G** and **E** are the cyclic prefix insertion and extraction matrices, respectively. The matrix **G** has dimensions (N+G)-by-N and **E** is of size N-by-(N+G+L–1). The superscript $(.)^{\mathrm{H}}$ states for Hermitian, i.e. transpose and conjugate; G is the cyclic prefix length, whereas L is the length of the channel impulse response calculated as *ceiling* $((\mathrm{T_m}/\mathrm{T})\cdot\mathrm{N})$ (in samples), where *ceiling* (*x*) stands for the nearest integer towards infinity of the real number *x*.

The matrices **G** and **E** are as follows,

$$\mathbf{G} = \begin{pmatrix} \mathbf{0}_{(\mathrm{G x (N - G)})} & \mathbf{I}_{(\mathrm{GxG})} \\ \mathbf{I}_{(\mathrm{N/2})\mathrm{x}(\mathrm{N/2})} & \mathbf{0}_{(\mathrm{N/2})\mathrm{x}(\mathrm{N/2})} \\ \mathbf{0}_{(\mathrm{N/2})\mathrm{x}(\mathrm{N/2})} & \mathbf{I}_{(\mathrm{N/2})\mathrm{x}(\mathrm{N/2})} \end{pmatrix}, \tag{C.2a}$$

$$\mathbf{E} = \begin{pmatrix} \mathbf{0}_{(\mathrm{NxG})} & \mathbf{I}_{(\mathrm{NxN})} & \mathbf{0}_{(\mathrm{Nx(L - 1)})} \end{pmatrix}, \tag{C.2b}$$

where **0** states for the null matrix, **I** is the identity matrix and the subscript indicates the matrix dimensions (rows x columns). The matrix **H** represents a convolution with the sequence $h = [h_0,\ldots,h_{\mathrm{L}-1}]^{\mathrm{T}}$ and is expressed as,

$$\mathbf{H}=\begin{pmatrix} h_0 & 0 & 0 & 0 & 0 & 0 & \cdots & & 0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & & \cdots & & & 0 & 0 \\ \vdots & h_1 & h_0 & 0 & 0 & \vdots & & & \cdots & \vdots & & 0 \\ h_{\mathrm{L}-1} & \vdots & h_1 & h_0 & 0 & 0 & & & & & \vdots & \\ 0 & h_{\mathrm{L}-1} & \vdots & & \ddots & 0 & & & & 0 & & \vdots \\ 0 & 0 & h_{\mathrm{L}-1} & & & & & & & 0 & 0 & \\ & 0 & 0 & \ddots & & & & & & & 0 & 0 \\ \vdots & & 0 & 0 & & & \ddots & \cdots & & & \ddots & 0 \\ & \vdots & & 0 & 0 & & & & & & & h_0 \\ 0 & & \vdots & & & & & & 0 & \ddots & & h_1 \\ 0 & 0 & & \vdots & & & \cdots & & 0 & 0 & h_{\mathrm{L}-1} & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & 0 & 0 & 0 & h_{\mathrm{L}-1} \end{pmatrix}. \tag{C.3}$$

The matrix **F** is of dimensions N-by-N and can be written as

$$\mathbf{F}=\begin{pmatrix} W_{0,0} & W_{0,1} & \cdots & W_{0,\mathrm{N}-1} \\ W_{1,0} & W_{1,1} & \cdots & W_{0,\mathrm{N}-1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{\mathrm{N}-1,0} & W_{\mathrm{N}-1,1} & \cdots & W_{\mathrm{N}-1,\mathrm{N}-1} \end{pmatrix}, \tag{C.4}$$

with $W_{n,k}=(\sqrt{\mathrm{N}})^{-1}\cdot e^{j\frac{2\pi}{\mathrm{N}}nk}$; $0 \le n, k \le \mathrm{N}-1$.

By analyzing (C.1) it is found that the product $\mathbf{G}\cdot\mathbf{F}$ yields a matrix, which can be expressed as

$$\mathbf{G}\cdot\mathbf{F}=\begin{pmatrix} \mathbf{F_G} \\ \mathbf{F} \end{pmatrix}, \tag{C.5}$$

where $\mathbf{F_G}$ is a G-by-N matrix with elements $W_{n,k}$; n = N–G, …, N–1; k = 0, …, N–1. Furthermore, the product $\mathbf{E}\cdot\mathbf{H}$ can be expressed as

$$\mathbf{E}\cdot\mathbf{H}=(\mathbf{H_1} \quad \mathbf{H_2}), \tag{C.6}$$

being $\mathbf{H_1}$ an N-by-G matrix and $\mathbf{H_2}$ an N-by-N matrix with the general form

$$\mathbf{H_1}=\begin{pmatrix} h_{\mathrm{G}} & h_{\mathrm{G}-1} & \cdots & h_1 \\ h_{\mathrm{G}+1} & h_{\mathrm{G}} & \cdots & h_2 \\ \vdots & \vdots & \ddots & \vdots \\ h_{\mathrm{G}+\mathrm{N}-1} & h_{\mathrm{G}+\mathrm{N}-2} & \cdots & h_{\mathrm{N}} \end{pmatrix}, \tag{C.7a}$$

$$\mathbf{H_2}=\begin{pmatrix} h_0 & 0 & \cdots & 0 \\ h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{\mathrm{N}-1} & h_{\mathrm{N}-2} & \cdots & h_0 \end{pmatrix}. \tag{C.7b}$$

Introducing (C.5) and (C.6) into (C.1), yields

$$\mathbf{A_R}=\mathbf{F^H}\cdot(\mathbf{H_1} \quad \mathbf{H_2})\cdot\begin{pmatrix} \mathbf{F_G} \\ \mathbf{F} \end{pmatrix}\cdot\mathbf{A}=\mathbf{F^H}\cdot(\mathbf{H_1}\cdot\mathbf{F_G}+\mathbf{H_2}\cdot\mathbf{F})\cdot\mathbf{A} \tag{C.8}$$

A further simplification can be achieved by considering that

$$\mathbf{F_G} = \begin{pmatrix} \mathbf{0}_{\mathrm{Gx(N-G)}} & \mathbf{I}_{\mathrm{GxG}} \end{pmatrix} \cdot \mathbf{F}, \tag{C.9}$$

and substituting into (C.8), yields

$$\mathbf{A_R} = \mathbf{F^H} \cdot \left(\mathbf{H_1} \cdot \begin{pmatrix} \mathbf{0}_{\mathrm{Gx(N-G)}} & \mathbf{I}_{\mathrm{GxG}} \end{pmatrix} + \mathbf{H_2} \right) \cdot \mathbf{F} \cdot \mathbf{A} = \mathbf{F^H} \cdot \mathbf{C} \cdot \mathbf{F} \cdot \mathbf{A}. \tag{C.10}$$

For the case of L ≤ G+1, the matrix **C** is a *circulant* matrix with the form

$$\mathbf{C} = \begin{pmatrix} h_0 & 0 & 0 & 0 & 0 & h_{\mathrm{L}-1} & \vdots & h_1 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & h_{\mathrm{L}-1} & \vdots \\ \vdots & h_1 & h_0 & 0 & 0 & 0 & 0 & h_{\mathrm{L}-1} \\ h_{\mathrm{L}-1} & \vdots & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & h_{\mathrm{L}-1} & \vdots & h_1 & h_0 & 0 & 0 & 0 \\ 0 & 0 & h_{\mathrm{L}-1} & \vdots & h_1 & h_0 & 0 & 0 \\ \vdots & \vdots & 0 & h_{\mathrm{L}-1} & \vdots & h_1 & h_0 & 0 \\ 0 & 0 & 0 & 0 & h_{\mathrm{L}-1} & \vdots & h_1 & h_0 \end{pmatrix}. \tag{C.11}$$

It can be demonstrated ([Sjoe96] and references therein) that **C** being a circulant matrix, **C** is diagonalized by the DFT matrix, i.e. $\mathbf{C} = \mathbf{F} \cdot \mathbf{\Sigma} \cdot \mathbf{F^H}$, where $\mathbf{\Sigma}$ is a diagonal matrix with the elements in the main diagonal being the DFT of the first column of **C**. This is equivalent to say that the eigenvalues of **C** are the Discrete Fourier Transform of the first column of **C**. Considering this theorem, equation (C.10) results as follows

$$\mathbf{A_R} = \begin{pmatrix} H_0 & 0 & 0 & 0 \\ 0 & H_1 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & H_{\mathrm{N}-1} \end{pmatrix} \cdot \mathbf{A}. \tag{C.12}$$

Hence, a correct cyclic extension helps in keeping the orthogonality inside the symbol, since each sample $A_{\mathrm{R}k} = H_k \cdot A_k$, and no crosstalk occurs. This property simplifies very much the channel equalization on the receiver side in comparison with the traditional single carrier transmissions, where an inverse convolution has to be carried out. On the contrary, when L > G+1 the matrix **C** is no longer a circulant matrix and the samples $A_{\mathrm{R}k}$ may be expressed as

$$A_{\mathrm{R}k} = H_k \cdot A_k + \sum_{j=0;\; j\neq k}^{\mathrm{N}-1} \gamma_j^k \cdot A_j \,, \tag{C.13}$$

with $0 \leq k \leq \mathrm{N}-1$, and where the second term stands for the ICI.

Note that this analysis has been made considering a scenario where a single OFDM symbol is transmitted and the cyclic prefix may be chosen to be one sample shorter than the channel impulse response. In a real scenario, however, this will not be the case and the cyclic prefix length (G) should be always bigger or equal to the channel impulse response (L) in order to avoid ISI.

Appendix D

ETSI Channel Models for the 5 GHz ISM Band

Tap #	Delay (ns)	Average Relative Power (dB)	Ricean K	Doppler Model
1	0	0.0	0	Jake's
2	10	-0.9	0	Jake's
3	20	-1.7	0	Jake's
4	30	-2.6	0	Jake's
5	40	-3.5	0	Jake's
6	50	-4.3	0	Jake's
7	60	-5.2	0	Jake's
8	70	-6.1	0	Jake's
9	80	-6.9	0	Jake's
10	90	-7.8	0	Jake's
11	110	-4.7	0	Jake's
12	140	-7.3	0	Jake's
13	170	-9.9	0	Jake's
14	200	-12.5	0	Jake's
15	240	-13.7	0	Jake's
16	290	-18.0	0	Jake's
17	340	-22.4	0	Jake's
18	390	-26.7	0	Jake's

Table D.1. Model A. The parameter K in the fourth column corresponds to K_2 in (2.7).

Tap #	Delay (ns)	Average Relative Power (dB)	Ricean K	Doppler Model
1	0	-2.6	0	Jake’s
2	10	-3.0	0	Jake’s
3	20	-3.5	0	Jake’s
4	30	-3.9	0	Jake’s
5	50	0.0	0	Jake’s
6	80	-1.3	0	Jake’s
7	110	-2.6	0	Jake’s
8	140	-3.9	0	Jake’s
9	180	-3.4	0	Jake’s
10	230	-5.6	0	Jake’s
11	280	-7.7	0	Jake’s
12	330	-9.9	0	Jake’s
13	380	-12.1	0	Jake’s
14	430	-14.3	0	Jake’s
15	490	-15.4	0	Jake’s
16	560	-18.4	0	Jake’s
17	640	-20.7	0	Jake’s
18	730	-24.6	0	Jake’s

Table D.2. Model B. The parameter K in the fourth column corresponds to K_2 in (2.7).

Tap #	Delay (ns)	Average Relative Power (dB)	Ricean K	Doppler Model
1	0	-3.3	0	Jake’s
2	10	-3.6	0	Jake’s
3	20	-3.9	0	Jake’s
4	30	-4.2	0	Jake’s
5	50	0.0	0	Jake’s
6	80	-0.9	0	Jake’s
7	110	-1.7	0	Jake’s
8	140	-2.6	0	Jake’s
9	180	-1.5	0	Jake’s
10	230	-3.0	0	Jake’s
11	280	-4.4	0	Jake’s
12	330	-5.9	0	Jake’s
13	400	-5.3	0	Jake’s
14	490	-7.9	0	Jake’s
15	600	-9.4	0	Jake’s
16	730	-13.2	0	Jake’s
17	880	-16.3	0	Jake’s
18	1,050	-21.2	0	Jake’s

Table D.3. Model C. The parameter K in the fourth column corresponds to K_2 in (2.7).

Tap #	Delay (ns)	Average Relative Power (dB)	Ricean K	Doppler Model
1	0	0.0	10	Jake’s + spike
2	10	-10.0	0	Jake’s
3	20	-10.3	0	Jake’s
4	30	-10.6	0	Jake’s
5	50	-6.4	0	Jake’s
6	80	-7.2	0	Jake’s
7	110	-8.1	0	Jake’s
8	140	-9.0	0	Jake’s
9	180	-10.8	0	Jake’s
10	230	-12.3	0	Jake’s
11	280	-11.7	0	Jake’s
12	330	-14.3	0	Jake’s
13	400	-15.8	0	Jake’s
14	490	-19.6	0	Jake’s
15	600	-22.7	0	Jake’s
16	730	-27.6	0	Jake’s
17	880	-16.3	0	Jake’s
18	1,050	-21.2	0	Jake’s

Table D.4. Model D. The parameter K in the fourth column corresponds to K_2 in (2.7).

Tap #	Delay (ns)	Average Relative Power (dB)	Ricean K	Doppler Model
1	0	0.0	0	Jake’s
2	10	-10.0	0	Jake’s
3	20	-10.3	0	Jake’s
4	40	-10.6	0	Jake’s
5	70	-6.4	0	Jake’s
6	100	-7.2	0	Jake’s
7	140	-8.1	0	Jake’s
8	190	-9.0	0	Jake’s
9	240	-10.8	0	Jake’s
10	320	-12.3	0	Jake’s
11	430	-11.7	0	Jake’s
12	560	-14.3	0	Jake’s
13	710	-15.8	0	Jake’s
14	880	-19.6	0	Jake’s
15	1,070	-22.7	0	Jake’s
16	1,280	-27.6	0	Jake’s
17	1,510	-16.3	0	Jake’s
18	1,760	-21.2	0	Jake’s

Table D.5. Model E. The parameter K in the fourth column corresponds to K_2 in (2.7).

Appendix E

SNR Degradation due to Timing Offsets

IN ORDER TO INVESTIGATE THE EFFECTS ON THE RECEIVED SEQUENCE when the condition in (3.21) is not fully satisfied, two possible cases are considered:

Case 1: $\theta > 0$

Under this condition, the only values of l and l' that make sense in the analysis are those for which $\Delta l = 0, -1$ (see Figure 3.3). Initially, the case $\Delta l = 0$ is analyzed, yielding the following expression for (3.17),

$$r^{\theta}_{n,l'} = \frac{1}{\sqrt{\mathrm{N}}} \sum_{i=1}^{\mathrm{P}} h_i(l') \cdot \left(\sum_{\substack{k=-\mathrm{N_u}/2 \\ (k \neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}k\left(n - \frac{\mathrm{N}}{\mathrm{T}}\tau_i + \theta\right)} \right) \cdot w\left(n + \theta - \tau_i \cdot (\mathrm{N}/\mathrm{T}) + \mathrm{N_G}\right) + v(n') \,. \tag{E.1}$$

The receiver will perform a DFT operation on $r^{\theta}_{n,l'}$ in (E.1), which results in the following expression for the received data

$$Z_{m,l'} = \frac{1}{\mathrm{N}} \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k \neq 0)}}^{\mathrm{N_u}/2} A_{k,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot\theta} \left(\sum_{i=1}^{\mathrm{P}} h_i(l) \cdot e^{-j\frac{2\pi}{\mathrm{T}}k\cdot\tau_i} \cdot W_i((k-m),\theta) \right) + v(n') \,, \tag{E.2}$$

with $-\mathrm{N_u}/2 \leq m \leq \mathrm{N_u}/2$ ($m \neq 0$) and

$$W_i((k-m),\theta) = \sum_{n=0}^{\mathrm{N}-1} e^{j\frac{2\pi}{\mathrm{N}}(k-m)n} w\left(n + \theta - \tau_i \cdot (\mathrm{N}/\mathrm{T}) + \mathrm{N_G}\right). \tag{E.3}$$

At this point, (E.3) has to be analyzed carefully considering the function $w(n)$, as given in (3.19). In general, (E.3) can be written as follows

$$W_i((k-m),\theta) = \sum_{n=n_{i,1}}^{n_{i,2}} e^{j\frac{2\pi}{\mathrm{N}}(k-m)n} = e^{j\frac{\pi}{\mathrm{N}}(k-m)(n_{i,1}+n_{i,2})} \frac{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)(n_{i,2}-n_{i,1}+1)\right)}{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)\right)}, \tag{E.4}$$

where the counting range in the summation $[n_{i,1}, n_{i,2}]$ depends on θ, τ_i and $\mathrm{N_G}$. The following cases may be distinguished (see Figure E.1):

1) $\theta > (\mathrm{N/T})\cdot\tau_i \Rightarrow n_{i,1} = 0; n_{i,2} = \mathrm{N} - 1 - \theta + (\mathrm{N/T})\cdot\tau_i$
2) $0 < \theta < -\mathrm{N_G} + (\mathrm{N/T})\cdot\tau_i \Rightarrow n_{i,1} = -\mathrm{N_G} - \theta + (\mathrm{N/T})\cdot\tau_i; n_{i,2} = \mathrm{N} - 1$. This situation will only occur if the channel impulse response is longer than the cyclic prefix, i.e. if there is a value *i* for which $(\mathrm{N/T})\cdot\tau_i > \mathrm{N_G}$.
3) otherwise $\Rightarrow n_{i,1} = 0; n_{i,2} = \mathrm{N} - 1$

Now it is possible to calculate the values $(n_{i,1} + n_{i,2})$ and $(n_{i,2} - n_{i,1} + 1)$ in (E.4) in the following compact form:

$$(n_{i,1} + n_{i,2}) = \mathrm{N} - 1 - \Delta\gamma_i , \tag{E.5a}$$

$$(n_{i,2} - n_{i,1} + 1) = \mathrm{N} - \Delta\varepsilon_i , \tag{E.5b}$$

where $\Delta\gamma_i$ and $\Delta\varepsilon_i$ are defined as

$$\Delta\varepsilon_i = \begin{cases} \theta - \tau_i \frac{\mathrm{N}}{\mathrm{T}}; & \theta > \tau_i \frac{\mathrm{N}}{\mathrm{T}} \\ \tau_i \frac{\mathrm{N}}{\mathrm{T}} - \mathrm{N_G} - \theta; & 0 < \theta < -\mathrm{N_G} + \tau_i \frac{\mathrm{N}}{\mathrm{T}} \\ 0; & \text{otherwise} \end{cases}, \tag{E.6a}$$

$$\Delta\gamma_i = \begin{cases} \theta - \tau_i \frac{\mathrm{N}}{\mathrm{T}}; & \theta > \tau_i \frac{\mathrm{N}}{\mathrm{T}} \\ \mathrm{N_G} + \theta - \tau_i \frac{\mathrm{N}}{\mathrm{T}}; & 0 < \theta < -\mathrm{N_G} + \tau_i \frac{\mathrm{N}}{\mathrm{T}} \\ 0; & \text{otherwise} \end{cases}, \tag{E.6b}$$

and (E.4) can be expressed as

$$W_i((k-m),\theta) = e^{j\frac{\pi}{\mathrm{N}}(k-m)(\mathrm{N}-1-\Delta\gamma_i)} \frac{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)(\mathrm{N}-\Delta\varepsilon_i)\right)}{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)\right)}. \tag{E.7}$$

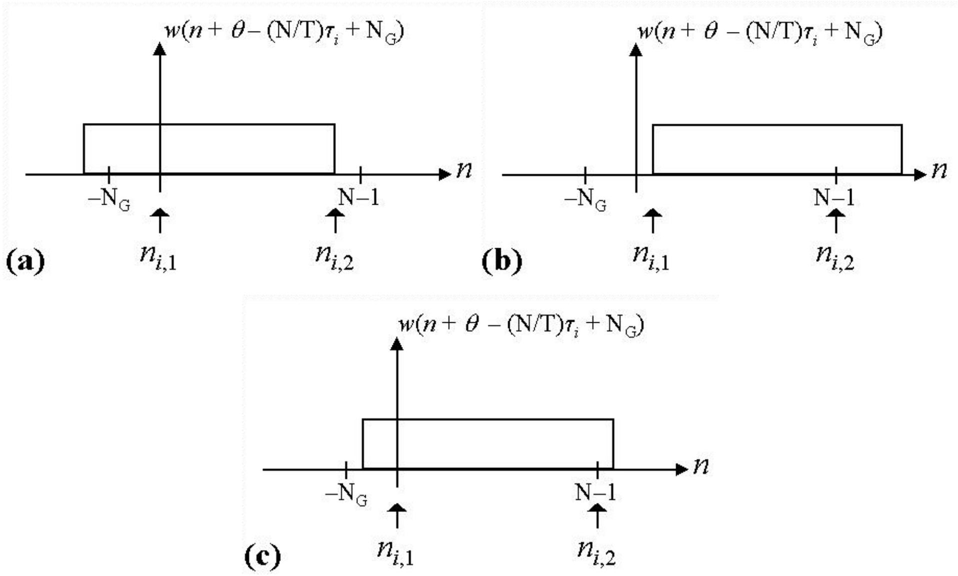


Figure E.1. Determination of the indices $n_{i,1}$ and $n_{i,2}$ in (E.4): a) $\theta > (\mathrm{N/T})\cdot\tau_i$; b) $0 < \theta < -\mathrm{N_G} + (\mathrm{N/T})\cdot\tau_i$; c) otherwise.

Considering the previous results, the following function is defined

$$\alpha((k-m),\theta)=\frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{P}} h_i(l)\cdot e^{-j\frac{2\pi}{\mathrm{T}}k\cdot\tau_i}\cdot W_i((k-m),\theta)\,, \tag{E.8}$$

with

$$\begin{aligned}\alpha(0,\theta)&=\sum_{i=1}^{\mathrm{P}} h_i(l)\cdot e^{-j\frac{2\pi}{\mathrm{T}}k\cdot\tau_i}\cdot\frac{\mathrm{N}-\Delta\varepsilon_i}{\mathrm{N}}\\ &=H_{k,l}-\frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{P}} h_i(l)\cdot\Delta\varepsilon_i\cdot e^{-j\frac{2\pi}{\mathrm{T}}k\cdot\tau_i}\\ &\equiv H_{k,l}-H_{k,l}^{\theta}\end{aligned} \tag{E.9}$$

and (E.2) results as follows

$$Z_{m,l'}=A_{m,l}\cdot e^{j\frac{2\pi}{\mathrm{N}}m\cdot\theta}\cdot\left(H_{m,l}-H_{m,l}^{\theta}\right)+\sum_{\substack{k=-\mathrm{N_u}/2;\\(k\neq m;k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l}\cdot\alpha((k-m),\theta)\cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot\theta}+v(n')\,, \tag{E.10}$$

with $-\mathrm{N_u}/2 \le m \le \mathrm{N_u}/2$ ($m \neq 0$).

In addition to this, there will be an ISI component added to expression (E.10) since some samples of the next symbol will be taken for integration (see Figure 3.3b). This can be analyzed by considering $\Delta l = -1$, i.e. $l=l'+1$. After manipulation of (3.17) and DFT calculation, the ISI component to be added to (E.10) results as follows,

$$Z_{m,l'}^{\mathrm{ISI}}\Big|_{\Delta l=-1}=\frac{1}{\mathrm{N}}\sum_{\substack{k=-\mathrm{N_u}/2;\\(k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l+1}\cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot\theta}\cdot e^{-j\frac{2\pi}{\mathrm{N}}k\cdot(\mathrm{N}+\mathrm{N_G})}\left(\sum_{i=1}^{\mathrm{P}} h_i(l+1)\cdot e^{-j\frac{2\pi}{\mathrm{T}}k\cdot\tau_i}\cdot W_{i,\Delta l=-1}^{\mathrm{ISI}}((k-m),\theta)\right), \tag{E.11}$$

with

$$W^{\mathrm{ISI}}_{i,\Delta l=-1}((k-m),\theta)=\sum_{n=0}^{\mathrm{N}-1} e^{j\frac{2\pi}{\mathrm{N}}(k-m)n}\, w\big(n+\theta-\mathrm{N}-\tau_i\cdot(\mathrm{N}/\mathrm{T})\big). \tag{E.12}$$

Again, considering the definition of $w(n)$ given in (3.19) and the expression in (E.4), (E.12) may be given by

$$W^{\mathrm{ISI}}_{i,\Delta l=-1}((k-m),\theta)=\begin{cases} e^{j\frac{2\pi}{\mathrm{N}}(k-m)(2\mathrm{N}-\theta-1+\tau_i\frac{\mathrm{N}}{\mathrm{T}})}\dfrac{\sin\left(\dfrac{\pi}{\mathrm{N}}(k-m)(\theta-\tau_i\dfrac{\mathrm{N}}{\mathrm{T}})\right)}{\sin\left(\dfrac{\pi}{\mathrm{N}}(k-m)\right)} & ;\ \theta>\tau_i\dfrac{\mathrm{N}}{\mathrm{T}} \\ 0 & ;\ \text{otherwise} \end{cases}. \tag{E.13}$$

In order to calculate the ICI and ISI powers in the presence of a fading channel and in the ideal case (AWGN only), it is considered that $0 < \theta < \tau_{\mathrm{M}}(\mathrm{N}/\mathrm{T}) < \tau_{\mathrm{P}}(\mathrm{N}/\mathrm{T})$ and the channel coefficients $h_i(l)$ are uncorrelated (uncorrelated scattering). Furthermore, the information between sub-carriers in one symbol and between different symbols is uncorrelated, but all sub-carriers have the same power σ_{A}^2. This results in the following power figures for the ICI and ISI terms, respectively,

$$\sigma^2_{\mathrm{ICI}}(m)=\sigma^2_{\mathrm{A}}\frac{1}{\mathrm{N}^2}\sum_{\substack{k=-\mathrm{N_u}/2\\(k\neq m;k\neq 0)}}^{\mathrm{N_u}/2}\sum_{i=1}^{\mathrm{M}}E\left\{\left|h_i(l)\right|^2\right\}\frac{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)(\mathrm{N}-\theta+\tau_i\dfrac{\mathrm{N}}{\mathrm{T}})\right)}{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)\right)}, \tag{E.14}$$

$$\sigma^2_{\mathrm{ISI}}(m)=\sigma^2_{\mathrm{A}}\frac{1}{\mathrm{N}^2}\sum_{\substack{k=-\mathrm{N_u}/2\\(k\neq 0)}}^{\mathrm{N_u}/2}\sum_{i=1}^{\mathrm{M}}E\left\{\left|h_i(l+1)\right|^2\right\}\frac{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)(\theta-\tau_i\dfrac{\mathrm{N}}{\mathrm{T}})\right)}{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)\right)}, \tag{E.15}$$

where the channel for symbol l and l+1 has a normalized power, i.e.

$$\sum_{i=1}^{\mathrm{P}}E\left\{\left|h_i(l)\right|^2\right\}=\sum_{i=1}^{\mathrm{P}}E\left\{\left|h_i(l+1)\right|^2\right\}=1. \tag{E.16}$$

The previous expressions are simplified when no channel is considered, yielding

$$\sigma^2_{\mathrm{ICI}}(m)=\sigma^2_{\mathrm{A}}\frac{1}{\mathrm{N}^2}\sum_{\substack{k=-\mathrm{N_u}/2\\(k\neq m;k\neq 0)}}^{\mathrm{N_u}/2}\frac{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)(\mathrm{N}-\theta)\right)}{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)\right)}, \tag{E.17}$$

$$\sigma^2_{\mathrm{ISI}}(m)=\sigma^2_{\mathrm{A}}\frac{1}{\mathrm{N}^2}\sum_{\substack{k=-\mathrm{N_u}/2\\(k\neq 0)}}^{\mathrm{N_u}/2}\frac{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)\cdot\theta\right)}{\sin^2\left(\dfrac{\pi}{\mathrm{N}}(k-m)\right)}. \tag{E.18}$$

In Figure 3.5 the Interference-to-Signal Ratio $(\sigma_{\mathrm{ICI}}^2 + \sigma_{\mathrm{ISI}}^2) / \sigma_{\mathrm{A}}^2$ is represented as a function of a positive timing offset for the case of an AWGN channel and a multipath channel model A with normalized power, as given in Table D.1.

<u>Case 2</u>: $\theta < \mathrm{N_m} - \mathrm{N_G}$ $(\theta < 0)$

In this case no ICI will occur since the orthogonality is preserved. Nevertheless, an ISI component will be present that can be obtained by considering $\Delta l = 1$. The general expression of the received data after DFT is given by

$$Z_{m,l'} = A_{m,l} \cdot H_{m,l} \cdot e^{j\frac{2\pi}{\mathrm{N}}m\cdot\theta} + Z^{\mathrm{ISI}}_{m,l'}\Big|_{\Delta l=+1} + v(n')\,, \tag{E.19}$$

with the ISI component being expressed as

$$Z^{\mathrm{ISI}}_{m,l'}\Big|_{\Delta l=+1} = \frac{1}{\mathrm{N}} \sum_{\substack{k=-\mathrm{N_u}/2 \\ (k\neq 0)}}^{\mathrm{N_u}/2} A_{k,l'-1} \cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot\theta} \cdot e^{j\frac{2\pi}{\mathrm{N}}k\cdot(\mathrm{N}+\mathrm{N_G})} \left(\sum_{i=1}^{\mathrm{P}} h_i(l'-1) \cdot e^{-j\frac{2\pi}{\mathrm{T}}k\cdot\tau_i} \cdot W^{\mathrm{ISI}}_{i,\Delta l=+1}((k-m),\theta) \right), \tag{E.20}$$

$-\mathrm{N_u}/2 \leq m \leq \mathrm{N_u}/2$ $(m \neq 0)$, and

$$W^{\mathrm{ISI}}_{i,\Delta l=+1}((k-m),\theta) = \sum_{n=0}^{\mathrm{N}-1} e^{j\frac{2\pi}{\mathrm{N}}(k-m)n}\, w\big(n + \theta + \mathrm{N} + 2\mathrm{N_G} - \tau_i \cdot (\mathrm{N}/\mathrm{T})\big). \tag{E.21}$$

Considering (E.4), (E.21) results in

$$W^{\mathrm{ISI}}_{i,\Delta l=+1}((k-m),\theta) = e^{j\frac{\pi}{\mathrm{N}}(k-m)\lambda}\, \frac{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)(\lambda+1)\right)}{\sin\left(\frac{\pi}{\mathrm{N}}(k-m)\right)}\,, \tag{E.22}$$

where λ stands for the minimum integer in the range $\lambda \in [0, \mathrm{P}-2]$ for which

$$\tau_{\mathrm{P}-\lambda-1} < (\theta + \mathrm{N_G}) \cdot (\mathrm{T/N}) < \tau_{\mathrm{P}-\lambda}\,. \tag{E.23}$$

Appendix F

Derivation of the Classical Maximum-Likelihood OFDM Synchronizer

THE PURPOSE OF THIS APPENDIX IS TO DETERMINE the expressions for the frequency and time discriminators in §4.2.1 making use of the classical Maximum-Likelihood approach in an AWGN scenario. Expressions (4.4) and (4.5) can be expressed in vector form as follows,

$$\mathbf{r} = \mathbf{s} + \mathbf{v}\,, \tag{F.1}$$

$$\mathbf{s} = (\mathbf{D} \circ \mathbf{F})\mathbf{A}\,, \tag{F.2}$$

where **D** and **F** are N-by-N square matrices with the following elements, respectively,

$$D_{n,k}(\varepsilon,\theta) = e^{j\frac{2\pi}{\mathrm{N}}\left(\varepsilon\cdot(n+\mathrm{N_G})-\theta\cdot k\right)} \cdot g\left(n-\theta+\mathrm{N_G}\right), \tag{F.3}$$

$$W_{n,k} = (\sqrt{\mathrm{N}})^{-1} \cdot e^{j\frac{2\pi}{\mathrm{N}}nk}\,, \tag{F.4}$$

and $0 \leq n,\ k \leq \mathrm{N}-1$. The symbol (○) denotes an element-by-element matrix multiplication. The vector **A** contains the transmitted sequence $\mathbf{A}=[A_0,\ldots,A_{\mathrm{N}-1}]^{\mathrm{T}}$ and **v** is a vector with samples from a zero-mean white Gaussian process.

In the Maximum Likelihood estimator, the parameter vector $\boldsymbol{\theta} = (\varepsilon,\ \theta)$ is obtained as

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}}\left(L(\mathbf{r};\boldsymbol{\theta})\right). \tag{F.5}$$

Since the vector $\boldsymbol{\theta}$ is supposed to be unknown but deterministic and **v** is Gaussian, the likelihood function is given by the following expression,

$$L(\mathbf{r};\boldsymbol{\theta}) = p(\mathbf{r};\boldsymbol{\theta}) = C \cdot e^{\frac{-1}{2\sigma^2}(\mathbf{r}-\mathbf{s})^{\mathrm{H}}(\mathbf{r}-\mathbf{s})} = C \cdot e^{\frac{-1}{2\sigma^2}(\mathbf{r}^{\mathrm{H}}\mathbf{r}+\mathbf{s}^{\mathrm{H}}\mathbf{s}-2\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\})}\,, \tag{F.6}$$

where the superscript ($^{\mathrm{H}}$) stands for the Hermitian transpose. A closer look to the previous expression shows that the term $\mathbf{r}^{\mathrm{H}}\mathbf{r}$ does not depend on $\boldsymbol{\theta}$. The term $\mathbf{s}^{\mathrm{H}}\mathbf{s}$ does not depend on ε, and the dependency on θ is smaller as N increases. This allows a simplification of (F.6), yielding

$$L(\mathbf{r};\boldsymbol{\theta}) = e^{\frac{1}{\sigma^2}\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\}}\,. \tag{F.7}$$

The previous expression for the likelihood function depends on the random data A_k, i.e. it is a random variable. In order to overcome this problem, an averaging over all the possible values of A_k is necessary. Nevertheless, taking the statistical expectation of (F.7) over A_k is mathematically intractable due to the exponential function. This is solved by considering a second order approach of the exponential (low SNR approximation), yielding

$$L(\mathbf{r};\boldsymbol{\theta}) = 1 + \frac{1}{\sigma^2}\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\} + \frac{1}{2\sigma^4}\left(\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\right)^2. \tag{F.8}$$

Averaging the previous expression and discarding the constant factor, which does not influence the final result, yields

$$L(\mathbf{r};\boldsymbol{\theta}) = \frac{1}{2\sigma^4}E_A\{(\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\})^2\}\,, \tag{F.9}$$

where $E_A\{.\}$ indicates the expectation over A_k. In the previous result it has been considered that $E\{A_k\} = 0$ and $E\{A_i \cdot A_j\} = 0$ for $i \neq j$.

In order to maximize (F.9), the derivative with respect to ε and θ has to be taken,

$$\frac{\partial}{\partial\varepsilon}L(\mathbf{r};\boldsymbol{\theta}) = \frac{1}{2\sigma^4}E_A\{2(\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\})\cdot\frac{\partial}{\partial\varepsilon}(\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\})\} = \frac{1}{\sigma^4}E_A\{\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\cdot\Re\mathrm{e}\{\frac{\partial}{\partial\varepsilon}\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\}\,, \tag{F.10a}$$

$$\frac{\partial}{\partial\theta}L(\mathbf{r};\boldsymbol{\theta}) = \frac{1}{2\sigma^4}E_A\{2(\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\})\cdot\frac{\partial}{\partial\theta}(\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\})\} = \frac{1}{\sigma^4}E_A\{\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\cdot\Re\mathrm{e}\{\frac{\partial}{\partial\theta}\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\}\,. \tag{F.10b}$$

Using λ as a dummy parameter, the expression inside the expectations in (F.10a/b) can be further simplified as follows,

$$\begin{aligned}
&E_A\{\Re\mathrm{e}\{\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\cdot\Re\mathrm{e}\{\frac{\partial}{\partial\lambda}\mathbf{s}^{\mathrm{H}}\mathbf{r}\}\} \\
&\qquad = E_A\{\Re\mathrm{e}\{((\mathbf{D}\circ\mathbf{F})\mathbf{A})^{\mathrm{H}}\mathbf{r}\}\cdot\Re\mathrm{e}\{\frac{\partial}{\partial\lambda}((\mathbf{D}\circ\mathbf{F})\mathbf{A})^{\mathrm{H}}\mathbf{r}\}\} \\
&\qquad = E_A\{\Re\mathrm{e}\{\mathbf{A}^{\mathrm{H}}(\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\}\cdot\Re\mathrm{e}\{\mathbf{A}^{\mathrm{H}}(\frac{\partial}{\partial\lambda}\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\}\} \\
&\qquad = E_A\Big\{\left(\Re\mathrm{e}\{\mathbf{A}^{\mathrm{H}}\}\Re\mathrm{e}\{(\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\} - \Im\mathrm{m}\{\mathbf{A}^{\mathrm{H}}\}\Im\mathrm{m}\{(\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\}\right) \\
&\qquad \times\left(\Re\mathrm{e}\{\mathbf{A}^{\mathrm{H}}\}\Re\mathrm{e}\{(\frac{\partial}{\partial\lambda}\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\} - \Im\mathrm{m}\{\mathbf{A}^{\mathrm{H}}\}\Im\mathrm{m}\{(\frac{\partial}{\partial\lambda}\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\}\right)\Big\}.
\end{aligned} \tag{F.11}$$

Now it is further considered that the real and imaginary parts of the vector **A** are independent and have zero mean, yielding

$$E_A\{\Re\mathrm{e}\{\mathbf{A}^{\mathrm{H}}\}\Im\mathrm{m}\{\mathbf{A}\}\} = E_A\{\Im\mathrm{m}\{\mathbf{A}^{\mathrm{H}}\}\Re\mathrm{e}\{\mathbf{A}\}\} = \mathbf{0}\,, \tag{F.12a}$$

$$E_A\{\Re\mathrm{e}\{\mathbf{A}^{\mathrm{H}}\}\Re\mathrm{e}\{\mathbf{A}\}\} = E_A\{\Im\mathrm{m}\{\mathbf{A}^{\mathrm{H}}\}\Im\mathrm{m}\{\mathbf{A}\}\} = \frac{\sigma_{\mathrm{A}}^2}{2}\mathbf{I}\,, \tag{F.12b}$$

and (F.10a/b) simplify as follows

$$\begin{aligned}\frac{\partial}{\partial\lambda}L(\mathbf{r};\boldsymbol{\theta}) &= \frac{\sigma_{\mathrm{A}}^2}{2\sigma^4}\Re\mathrm{e}\{\mathbf{r}^{\mathrm{H}}(\frac{\partial}{\partial\lambda}\mathbf{D}\circ\mathbf{F})(\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\}\\ &= \frac{\sigma_{\mathrm{A}}^2}{2\sigma^4}\Re\mathrm{e}\{\mathbf{r}^{\mathrm{H}}(\mathbf{D}_{\lambda}\circ\mathbf{F})(\mathbf{D}\circ\mathbf{F})^{\mathrm{H}}\mathbf{r}\}\\ &= C\cdot\Re\mathrm{e}\{((\mathbf{F}^{\mathrm{H}}\circ\mathbf{D}_{\lambda}^{\mathrm{H}})\mathbf{r})^{\mathrm{H}}(\mathbf{F}^{\mathrm{H}}\circ\mathbf{D}^{\mathrm{H}})\mathbf{r}\}\,,\end{aligned} \tag{F.13}$$

where $\mathbf{F}^{\mathrm{H}}$ is the DFT matrix and $\mathbf{D}_{\lambda} = (\partial\mathbf{D}/\partial\lambda)$.

The elements in the matrices $\mathbf{D}_{\varepsilon}$ and $\mathbf{D}_{\theta}$ are obtained from (F.3) as follows

$$D_{n,k}^{\varepsilon}(\varepsilon,\theta) = \frac{\partial}{\partial\varepsilon}D_{n,k}(\varepsilon,\theta) = j\frac{2\pi}{\mathrm{N}}(n+\mathrm{N_G})e^{j\frac{2\pi}{\mathrm{N}}(\varepsilon\cdot(n+\mathrm{N_G})-\theta\cdot k)}\cdot g\left(n-\theta+\mathrm{N_G}\right), \tag{F.14a}$$

$$D_{n,k}^{\theta}(\varepsilon,\theta) = \frac{\partial}{\partial\theta}D_{n,k}(\varepsilon,\theta) = e^{j\frac{2\pi}{\mathrm{N}}(\varepsilon\cdot(n+\mathrm{N_G})-\theta\cdot k)}\cdot\left(\frac{\partial}{\partial\theta}g(n-\theta+\mathrm{N_G}) - j\frac{2\pi}{\mathrm{N}}k\cdot g(n-\theta+\mathrm{N_G})\right). \tag{F.14b}$$

Particularizing (F.3) and (F.14a/b) for the steady state situation, i.e. $(\varepsilon,\theta)\to 0$, yields

$$D_{n,k}(\varepsilon,\theta)\Big|_{(\varepsilon,\theta)\to 0} = g(n+\mathrm{N_G})\,, \tag{F.15a}$$

$$D_{n,k}^{\varepsilon}(\varepsilon,\theta)\Big|_{(\varepsilon,\theta)\to 0} = j\frac{2\pi}{\mathrm{N}}(n+\mathrm{N_G})\cdot g(n+\mathrm{N_G})\,, \tag{F.15b}$$

$$D_{n,k}^{\theta}(\varepsilon,\theta)\Big|_{(\varepsilon,\theta)\to 0} = -j\frac{2\pi}{\mathrm{N}}k\cdot g(n+\mathrm{N_G}) - \frac{\partial}{\partial n}g(n+\mathrm{N_G})\,, \tag{F.15c}$$

with $0 \le n, k \le \mathrm{N}-1$.

The frequency and time error discriminators are then defined as

$$\mathrm{e}_{\varepsilon} = \frac{\partial}{\partial\varepsilon}L(\mathbf{r};\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}\to\mathbf{0}}\,, \tag{F.16a}$$

$$\mathrm{e}_{\theta} = \frac{\partial}{\partial\theta}L(\mathbf{r};\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}\to\mathbf{0}}\,. \tag{F.16b}$$

We should note at this point that the derivative of the conforming pulse appearing in (F.15c) does not allow the use of the rectangular pulse. Hence, a more general Root-Raised Cosine (RRC) pulse has to be considered which is defined as

$$g(t)=\begin{cases}\sqrt{\dfrac{1}{\mathrm{T}}} & ;\,|t|\le\dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\\ \sqrt{\dfrac{1}{2\mathrm{T}}\left\{1+\cos\left(\dfrac{\pi}{\alpha_{\mathrm{r}}}\left(\dfrac{|t|}{\mathrm{T}}-\dfrac{(1-\alpha_{\mathrm{r}})}{2}\right)\right)\right\}} & ;\,\dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\le|t|\le\dfrac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})\\ 0 & ;\,\text{otherwise}\end{cases}. \tag{F.17}$$

The expression given in (F.17) for $g(t)$ is equivalent to $w_{\mathrm{RC}}(t)$ in (2.29). Furthermore, the pulse $|g(t)|^2$ complies with the orthogonality condition established in (2.27). Interestingly, the pulse $g'(t)$ defined as

$$g'(t)=g(t)\cdot\frac{\partial g(t)}{\partial t}=\begin{cases}\dfrac{\pi}{2\mathrm{T}\alpha_{\mathrm{r}}}\sin\left(\dfrac{\pi}{\alpha_{\mathrm{r}}}\left(\dfrac{t}{\mathrm{T}}-\dfrac{(1-\alpha_{\mathrm{r}})}{2}\right)\right) & ;\,-\dfrac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})\le t\le-\dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\\ \dfrac{-\pi}{2\mathrm{T}\alpha_{\mathrm{r}}}\sin\left(\dfrac{\pi}{\alpha_{\mathrm{r}}}\left(\dfrac{t}{\mathrm{T}}-\dfrac{(1-\alpha_{\mathrm{r}})}{2}\right)\right) & ;\,\dfrac{\mathrm{T}}{2}(1-\alpha_{\mathrm{r}})\le t\le\dfrac{\mathrm{T}}{2}(1+\alpha_{\mathrm{r}})\\ 0 & ;\,\text{otherwise}\end{cases} \tag{F.18}$$

also complies with the condition in (2.27). The direct implication of this fact is that the timing estimations will not be affected by ICI, i.e. the estimator will not suffer from *self-noise*. The pulses $g(t)$ and $g'(t)$ are shown in a normalized form in Figure F.1 for a roll-off factor of $\alpha_{\mathrm{r}} = 0.25$. A block diagram of the resulting scheme is depicted in Figure 4.1. Furthermore, the s-curves for the discriminators in (F.16a/b) are obtained as $E\{\mathrm{e}_{\varepsilon}\}$, $E\{\mathrm{e}_{\theta}\}$ and are depicted in Figure 4.2a/b, respectively.

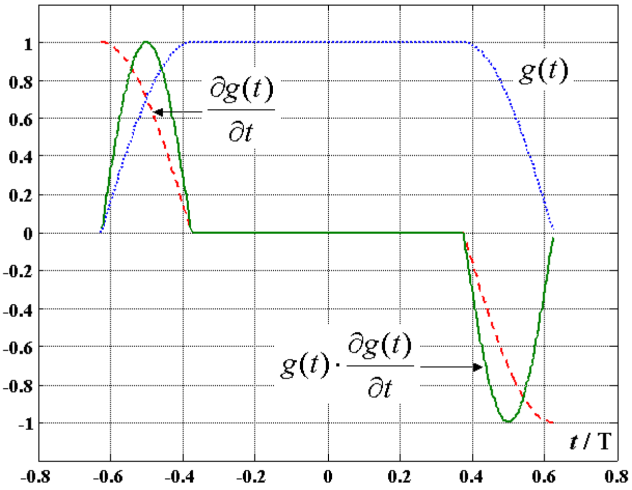


Figure F.1. The RRC pulse and its derivative ($\alpha_{\mathrm{r}} = 0.25$).

Bibliography

[Bahai99] Bahai, Ahmad R. S.; Saltzberg, Burton R.; *Multi-Carrier Digital Communications: Theory and Applications of OFDM*, Kluwer Academic / Plenum Publishers, 1999.

[Ban00] Paolo Banelli and Saverio Cacopardi, "Theoretical Analysis and Performance of OFDM Signals in Nonlinear AWGN Channels", IEEE Transactions on Communications, vol. 48, no. 3, pp. 430 - 441, March 2000.

[Beek95a] Jan-Jaap van de Beek, Ove Edfors, Magnus Sandell, Sarah Kate Wilson, Per Ola Börjesson, "On Channel Estimation in OFDM Systems", Proceedings of the IEEE Vehicular Technology Conference (VTC'95), vol. 2, pp. 815-819, Chicago (USA), July 1995.

[Beek95b] J.J. van de Beek, M. Sandell, M. Isaksson, P.O. Börjesson, "Low-Complex Frame Synchronization in OFDM Systems", Proceedings of the International Conference on Universal Personal Communications ICUPC'95, pp. 982-986, November 1995.

[Beek97] J.J. van de Beek, M. Sandell, P.O. Börjesson, "ML Estimation of Timing and Frequency Offset in OFDM Systems", IEEE Transactions on Signal Processing, vol. 45, no. 7, pp. 1800 - 1805, July 1997.

[Ben96] Maria-Gabriella di Benedetto, Paolo Mandarini, "Analysis of the Effect of the I/Q Base-Band Filter Mismatch in an OFDM Modem", ACTS Mobile Telecommunications Summit, Granada (Spain), November 1996 .

[Bing90] John A. C. Bingham, "Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come", IEEE Communications Magazine, pp. 5 - 14, May 1990.

[Bla03] Raúl Blázquez, private communication, Orlando, October 2003.

[Bron74] Bronstein, I.N.; Semendjajew, K.A.; *Taschenbuch der Mathematik*, 13. Auflage, BSB B.G. Teubner Verlagsgesellschaft, Leipzig, 1974 (in German).

[Brue99] K. Brüninghaus, H. Rohling, "Decision Directed Channel Estimation in Wireless OFDM Systems", Proc. of the 1st Int. OFDM-Workshop, pp. 14.1 - 14.4, Hamburg (Germany), Sept. 1999.

[Chiu00] Yun Chiu et al., *OFDM Receiver Design*, Final Report 12/12/2000. Downloaded from http://bwrc.eecs.berkeley.edu/People/Grad_Students/dejan/ee225c/ofdm.pdf

[Cim85] Leonard J. Cimini, "Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing", IEEE Trans. on Communications, vol. 33, no. 7, pp. 665-675, July 1985.

[Cim00] Leonard J. Cimini, Jr. and Nelson R. Sollenberger, "Peak-to-Average Power Ratio Reduction of an OFDM Signal Using Partial Transmit Sequences", IEEE Communication Letters, vol. 4, no. 3, pp. 86 - 88, March 2000.

[Class94] Ferdinand Classen, Heinrich Meyr, "Frequency Synchronization Algorithms for OFDM Systems suitable for Communication over Frequency Selective Fading Channels", Proceedings of the VTC'94, pp. 1655-1659, Stockholm (Sweden), June 1994.

[Coers01] E. Coersmeier, Y. Xu, L. Schwoerer, "High Precision Analog Front-End Transceiver Architecture for Wireless Local Area Network", in Proceedings of the 6th Int. OFDM-Workshop (InOWo) 2001, pp. 29.1-29.4, Hamburg (Germany), September 2001.

[Cool65] J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. Computation, vol. 19, pp. 297 - 301, 1965.

[Costa99] Elena Costa, Michele Midrio, and Silvano Pupolin, "Impact of Amplifier Nonlinearities on OFDM Transmission System Performance", IEEE Communication Letters, vol. 3, no. 2, pp. 37 - 39, February 1999.

[DAB] www.worlddab.org

[Daf93] Flavio Daffara, Antoine Chouly, "Maximum Likelihood Frequency Detectors for Orthogonal Multicarrier Systems", Proceedings of the IEEE ICC'93, pp. 766-771, Geneva (Switzerland), May 1993.

[Daf95] Flavio Daffara and Ottavio Adami, "A New Frequency Detector for Orthogonal Multicarrier Transmission Techniques", Proceedings of the IEEE VTC'95, vol. 2, pp. 804-809, Chicago (USA), July 1995.

[Dawid99] H. Dawid, H. Meyr, "CORDIC Algorithms and Architectures" in Chapter 22 of *Digital Signal Processing for Multimedia Systems*, K. J. Ray Liu (editor), Marcel Dekker Inc., 1999.

[DVB] www.dvb.org

[Edf96] Ove Edfors, Magnus Sandell, Jan-Jaap van de Beek, Sarah Kate Wilson, Per Ola Börjesson, "Analysis of DFT-based channel estimators for OFDM", Research Report TULEA 1996:17, Div. of Signal Processing, Luleå University of Technology, Sweden, 1996.

[Edf98] Ove Edfors, Magnus Sandell, Jan-Jaap van de Beek, Sarah Kate Wilson, Per Ola Börjesson, "OFDM Channel Estimation by Singular Value Decomposition", IEEE Transactions on Communications, vol. 46, no. 7, pp. 931-939, July 1998.

[Erup93] Lars Erup, F.M. Gardner, R.A. Harris, "Interpolation in Digital Modems – Part II: Implementation and Performance", IEEE Transactions on Communications, vol. 41, no. 6, pp. 998 - 1008, June 1993.

[ETSIa] ETSI DTS/BRAN 030003-1, "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2 Functional Specifications. Part 1 – Physical (PHY) layer; June 1999".

[ETSIb] ETSI, “Criteria for comparison”, Tech. Report 30701F, BRAN WG3 PHY Subgroup, May 1998

[Far88] C.W. Farrow, “A Continuously Variable Digital Delay Element”, Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 2641-2645, Espoo (Finland), June 1988.

[Fech99] Stefan A. Fechtel, Alfred Blaickner, “Efficient FFT and Equalizer Implementation for OFDM Receivers”, IEEE Trans. on Consumer Electronics, vol.45, no. 4, pp. 1104-1107, Nov. 1999.

[Flem03] Chip Fleming, “A Tutorial on Convolutional Coding with Viterbi Decoding”, downloaded from http://pw1.netcom.com/~chip.f/viterbi/tutorial.html

[For98] G. J. Forschini, M. J. Gans, “On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas”, Wireless Personal Communications, Kluwer Academic Publishers, vol. 6, no. 3, pp. 311-335, 1998.

[Fuji03] Toshio Fujisawa et al., “A Single-Chip 802.11a MAC/PHY with a 32-b RISC Processor ”, IEEE Journal of Solid-State Circuits, vol. 38, no. 11, pp. 2001-2009, November 2003.

[Gar93] Floyd M. Gardner, “Interpolation in Digital Modems – Part I: Fundamentals”, IEEE Transactions on Communications, vol. 41, no. 3, pp. 501 - 507, March 1993.

[Gol96] Golub, Gene H.; van Loan, Charles F.; *Matrix Computations*, The Johns Hopkins University Press, 3rd Edition, 1996.

[Grass01] E. Grass, K. Tittelbach-Helmrich, U. Jagdhold, A. Troya, G. Lippert, O. Krüger, J. Lehmann, K. Maharatna, K. F. Dombrowski, N. Fiebig, R. Kraemer, P. Mähönen., “On the Single-Chip Implementation of a Hiperlan/2 and IEEE 802.11a Capable Modem”, IEEE Personal Communications Magazine, vol. 8, no. 6, pp. 48 - 57, December 2001.

[Hay02] Haykin, S.S.; *Adaptive Filter Theory*, 4th Edition, Prentice Hall International Editions, 2002.

[Hend96] Henry Hendrix, “Viterbi Decoding Techniques in the TMS320C54x Family”, Texas Instruments' internal report SPRA071, June 1996.

[Hen00] Nabil Hentati and Marc Schrader, “Additive Algorithm for Reduction of Crestfactor (AARC)”, Proceedings of the 5th Int. OFDM-Workshop (InOWo) 2000, pp. 27.1 – 27.5, Hamburg (Germany), September 2000.

[Hoe91] P. Hoeher, “TCM on Frequency-Selective Land-Mobile Fading Channels”, Proceedings of the 5th Tirrenia Int. Workshop on Digital Comm., pp. 317-328, Tirrenia (Italy), September 1991.

[Hoe92] P. Hoeher, “A Statistical Discrete-Time Model for the WSSUS Multipath Channel”, IEEE Trans. Vehicular Tech., vol. 41, no. 4, pp. 461-468, Nov. 1992.

[Hoe97] Peter Hoeher, Stefan Kaiser, Patrick Robertson, “Two-Dimensional Pilot-Symbol-Aided Channel Estimation by Wiener Filtering”, Proceedings of the IEEE ICASSP'97, pp. 1845-1848, Munich (Germany), April 1997.

[Hsieh99] Meng-Han Hsieh, Che-Ho Wei, “A Low-Complexity Frame Synchronization and Frequency Offset Compensation Scheme for OFDM Systems over Fading Channels”, IEEE Transactions on Vehicular Technology, vol. 48, no. 5, pp. 1596-1609, September 1999.

[IEEE] IEEE P802.11a/D7.0, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High Speed Physical Layer in the 5 GHz Band”; July 1999.

[IEEE15] www.ieee802.org/15

[Jakes00] Jakes, W.C.; *Microwave Mobile Communications*, IEEE Press, 2000.

[Jeru92] Jeruchim, Michel C.; Balaban, Philip; Shanmugan, K. Sam; *Simulation of Communication Systems*, Plenum Press, New York, 1992.

[Johan96] H. Johansson, L. Wanhammar, “Design of Bireciprocal Linear-Phase Lattice Wave Digital Filters”, Report LiTH-ISY-R-1877, Linköping 1996.

[Johan97] H. Johansson, L. Wanhammar, “Design of Linear-Phase Lattice Wave Digital Filters”, Report LiTH-ISY-R-1930, Linköping 1997.

[Kai98] Kaiser, S.; *Multi-Carrier CDMA Mobile Radio Systems – Analysis and Optimization of Detection, Decoding and Channel Estimation*, Ph.D. Thesis, VDI Verlag, Reihe 10, Nr. 531, 1998.

[Kang99] H. W. Kang, Y. S. Cho, D. H. Youn, “On Compensating Nonlinear Distortions of an OFDM System Using an Efficient Adaptive Predistorter”, IEEE Transactions on Communications, vol. 47, no. 4, pp. 522 – 526, April 1999.

[Kay93] Kay, Steven M.; *Fundamentals of Statistical Signal Processing: Estimation Theory*; Prentice Hall, 1993.

[Kel96] T, Keller, L. Hanzo, “Orthogonal Frequency Division Multiplex Synchronization Techniques for Wireless Local Area Networks”, Proceedings of the International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'96), pp. 963-967, Taipei (Taiwan), 1996.

[Kim97] D. Kim, M.J. Narasimha, D.C. Cox, “Design of Optimal Interpolation Filter for Symbol Timing Recovery”, IEEE Transactions on Communications, vol. 45, no. 7, pp. 877-884, July 1997.

[Klatt03] Jörg Klatt, private communication, IHP, November 2003.

[Kothe03] René Kothe, private communication, IHP, October 2003.

[Krstic03a] M. Krstić, A. Troya, K. Maharatna, “Optimized Low-power Synchronizer Design for the IEEE 802.11a Standard”, Proceedings of the IEEE ICASSP'03, vol. II, pp. 333-336, Hong Kong (P.R. China), April 2003.

[Krstic03b] M. Krstić, K. Maharatna, A. Troya, E. Grass, U. Jagdhold, “Implementation of an IEEE 802.11a Compliant Low-Power Baseband Processor”, in Proceedings of the Intl. Conf. on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS 2003), vol. 1, pp. 97–100, Niš (Yugoslavia), October 2003.

[Krstic03c] Miloš Krstić, private communication, IHP, November 2003.

[Kruk99] A. Krukowski, I. Kale, “Almost Linear-Phase Polyphase IIR Lowpass/Highpass Filter Approach”, Proceedings of the ISSPA'99, Brisbane (Australia), August 1999.

[Kruk01] A. Krukowski, I. Kale, “Polyphase Filter Design with Reduced Phase Non-Linearity”, Proceedings of the 5th WSES/IEEE World Multiconference on Circuits, Systems, Communications & Computers CSCC'01, Rethymnon, Crete (Greece), July 2001.

[Laak96] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen and Unto K. Laine, “Splitting the Unit Delay: Tools for fractional delay filter design”, IEEE Signal Processing Magazine, pp. 30-60, January 1996.

[Land99] Daniel Landström, Sarah Kate Wilson, Jan-Jaap van de Beek, Per Ödling, Per Ola Börjesson, “Symbol Time Offset Estimation in Coherent OFDM Systems”, Proceedings of the IEEE

International Conference on Communications ICC'99, vol. 1, pp. 500-505, Vancouver (Canada), June 1999.

[Law97] S.S. Lawson, “On design techniques for approximately linear phase recursive digital filters”, Proceedings of the ISCAS'97, pp. 2212-2215, June 1997.

[Maha00] Maharatna, K.; *CORDIC Based Signal Processing for Biomedical Application*, Ph.D. Thesis, Department of Physics, Jadavpur University, Calcutta (India), 2000.

[Maha03a] K. Maharatna, E. Grass and U. Jagdhold, “A Novel 64-point FFT/IFFT Processor for IEEE 802.11a Standard”, Proceedings of the IEEE ICASSP'03, vol. II, pp. 321-324, April 2003.

[Maha03b] Dr. Koushik Maharatna, private communication, IHP, April 2003.

[Maha04a] K. Maharatna, A. Troya, S. Banerjee, E. Grass, M. Krstić, “A 16-Bit Low-Power CORDIC Rotator for High-Speed LAN”, submitted to IEEE PIMRC'04, Barcelona (Spain), September 2004.

[Maha04b] K. Maharatna, A. Troya, M. Krstić, E. Grass, U. Jagdhold, “A CORDIC Like Processor for Computation of Arctangent and Absolute Magnitude of a Vector”, accepted for publication at IEEE ISCAS '04, Vancouver (Canada), May 2004.

[Mari95] Mariño Acebal, J.B.; Vallverdú i Bayes, F.; Rodríguez Fonollosa, J.A.; Moreno Bilbao, A.; *Tractament digital del senyal: una introducció experimental*, (Catalan version), CPET-UPC, 1995.

[Mar01] Martrat Sotil, José; *Sincronización No Asistida en Sistemas Multiportadora xDSL y Power Line*, M.Sc. Thesis (in Spanish), Polytechnical University of Catalonia (UPC), Barcelona (Spain), 2001.

[Meyr97] Meyr, H.; Moeneclaey, M.; Fechtel, S.; *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, John Wiley & Sons, 1998.

[Mig96] Vittoria Mignone, Alberto Morello, “CD3-OFDM: A Novel Demodulation Scheme for Fixed and Mobile Receivers”, IEEE Transactions on Communications, vol. 44, no. 9, pp. 1144-1151, September 1996.

[Moon00] J.K. Moon, S.I. Choi, “Performance of Channel Estimation Methods for OFDM Systems in Multipath Fading Channels”, IEEE Transactions on Consumer Electronics, vol. 46, no. 1, pp. 161-170, February 2000.

[Moose94] Paul H. Moose, “A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction”, IEEE Transactions on Communications, vol. 42, no. 10, pp. 2908-2914, October 1994.

[Mor99] Michele Morelli and Umberto Mengalli, “An Improved Frequency Offset Estimator for OFDM Applications”, IEEE Communications Letter, vol. 3, no. 3, pp. 75-77, March 1999.

[Muel98] Stefan H. Müller-Weinfurtner, “On the Optimality of Metrics for Coarse Frame Synchronization in OFDM: A Comparison”, Proceedings IEEE Int. Symp. On Personal, Indoor and Mobile Radio Communications, pp. 533-537, Boston, MA (USA), September 1998.

[Mus95] Claus Muschallik, “Influence of RF Oscillators on an OFDM Signal”, IEEE Transactions on Consumer Electronics, vol. 41, no. 3, pp. 592 - 603, August 1995.

[Nee00] van Nee, Richard; Prasad, Ramjee; *OFDM for Wireless Multimedia Communications*, Artech House Publishers, 2000.

[Negi98] Rohit Negi, John Cioffi, “Pilot Tone Selection for Channel Estimation in a Mobile OFDM System”, IEEE Transactions on Consumer Electronics, vol. 44, no. 3, pp. 1122-1128, August 1998.

[Opp99] Oppenheim, A. V.; Schafer, R. W.; *Discrete-Time Signal Processing*, 2nd Edition, Prentice-Hall International, 1999.

[Pan03] Goran Panić, private communication, IHP, November 2003.

[Pir98] P. Pirsch, *Architectures for Digital Signal Processing*, Wiley, 1998.

[Pro95] Proakis, John G.; *Digital Communications*, 3rd ed., McGraw-Hill International Editions, 1995.

[Pro96] Proakis, John G.; Manolakis, Dimitris G.; *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice Hall, 3rd Edition, 1996.

[Rob95] P. Robertson and S. Kaiser, “Analysis of the Effects of Phase-Noise in Orthogonal Frequency Division Multiplex (OFDM) Systems”, Proceedings of the IEEE ICC’95, pp. 1652 – 1657, June 1995.

[Sand95] S. D. Sandberg, M. A. Tzannes, “Overlapped Discrete Multitone Modulation for High Speed Copper Wire Communications”, Journal on Selected Areas in Communications, vol. 13, no. 9, pp. 1571 - 1585, December 1995.

[Sandell96] Magnus Sandell, Ove Edfors, “A comparative study of pilot-based channel estimators for wireless OFDM”, Research Report TULEA 1996:19, Div. of Signal Processing, Luleå University of Technology, Sweden, September 1996.

[Schmidl97] Timothy M. Schmidl and Donald C. Cox, “Robust Frequency and Timing Synchronization for OFDM”, IEEE Transactions on Communications, vol. 45, no. 12, pp. 1613-1621, December 1997.

[Schmidt00] H. Schmidt, K.-D. Kammeyer, “Impulse truncation for wireless OFDM systems”, Proceedings of the 5th Int. OFDM-Workshop 2000, pp. 34.1-34.5, Hamburg (Germany), September 2000.

[Schmidt01] Heiko Schmidt, Volker Kühn, Karl-Dirk Kammeyer, Reinhard Rückriem, Stefan Fechtel, “Channel Tracking in Wireless OFDM Systems”, Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando (USA), July 2001.

[Schwoe01] Ludwig Schwörer and Holger Wirz, “VLSI Implementation of IEEE 802.11a Physical Layer”, Proceedings of the 6th Intl. OFDM-Workshop (InOWo) 2001, pp. 28.1-28.4, Hamburg (Germany), September 2001.

[Sjoe96] Sjöström, Eva; *Singular Value Decomposition for Toeplitz Matrices*, Licentiate thesis, Linköping University, Sweden, May 1996.

[Speth97] Michael Speth, Ferdinand Classen, Heinrich Meyr, “Frame Synchronization of OFDM Systems in Frequency Selective Fading Channels”, Proceedings of the 47th IEEE VTC'97, vol. 3, pp. 1807-1811, Phoenix, AR (USA), May 1997.

[Speth98] Michael Speth, Dirk Daecke, Heinrich Meyr, “Minimum Overhead Burst Synchronization for OFDM based Broadband Transmission”, Proceedings of the IEEE Global Telecommunications Conference (Globecom'98), vol. 5, pp. 2777-2782, 1998.

[Speth99] M. Speth, S. A. Fechtel, H. Meyr, “Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM - Part I”, IEEE Transactions on Communications, vol. 47, no. 11, pp. 1668-1677, November 1999.

[SPW] Cadence's SPW Block Reference Manual.

[Stant98] Branimir Stantchev and Gerhard Fettweis, “Burst Synchronization for OFDM-based Cellular Systems with Separate Signaling Channel”, Proceedings of the IEEE Vehicular Technology Conference, pp. 758-762, Ottawa (Canada), May 1998.

[Tar00] Vahid Tarokh and Hamid Jafarkhani, “On the Computation and Reduction of the Peak-to-Average Power Ratio in Multicarrier Communications”, IEEE Transactions on Communications, vol. 48, no. 1, pp. 37 – 44, January 2000.

[Tel95] I. E. Telatar, “Capacity of Multi-Antenna Gaussian Channels”, AT&T Bell Laboratories internal Technical Memorandum, 1995.

[Thoen02] S. Thoen, L. Van der Perre, M. Engels, “Channel Time-Variance for Fixed Wireless Communications: Modeling and Impact”, Proceedings of the IASTED International Conference on Wireless and Optical Communications, pp. 239 - 244, July 17-19, 2002.

[Thom02] John Thomson et al., “An Integrated 802.11a Baseband and MAC Processor”, in Digest of Technical Papers of the IEEE ISSCC 2002, vol. 1, pp. 126-127, February 2002.

[Tos02] F. Tosato, P. Bisaglia, “Simplified Soft-Output Demapper for Binary Interleaved COFDM with Application to HIPERLAN/2”, in Proceedings of the IEEE ICC 2002, vol. 2, pp. 664–668, April 2002.

[Troya99] Troya, Alfonso; *Design and Simulation of the RF Interface for an HFC Network Monitoring Device*, M.Sc. Thesis, UPC, Barcelona (Spain), July 1999.

[Troya01] Alfonso Troya, “On the Performance of a Decision-Directed Channel Estimator for OFDM”, Proceedings of the 6th International OFDM-Workshop (InOWo'01), pp. 10.1-10.5, Hamburg (Germany), September 2001.

[Troya02] A. Troya, K. Maharatna, M. Krstic, “OFDM Synchronizer Implementation for an IEEE 802.11a Compliant Modem”, Proceedings of the IASTED International Conference on Wireless and Optical Communications (WOC'02), pp. 152-157, Banff (Canada), July 2002.

[Troya03] A. Troya, M. Krstic, K. Maharatna, “Simplified Residual Phase Correction Mechanism for the IEEE 802.11a Standard”, Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall) 2003, Orlando (USA), October 2003.

[Vaid93] Vaidyanathan, P.P.; *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[Vit67] Andrew J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”, IEEE Transactions on Information Theory, vol. 13, no. 2, pp. 260-269, April 1967.

[Vold59] J. E. Volder, “The CORDIC trigonometric computing technique”, IRE Trans. Electronic Computers, vol. EC-8, pp. 330 - 334, September 1959.

[Wein71] S. B. Weinstein, P. M. Ebert, “Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform”, IEEE Trans. Comm. Tech., vol. 19, pp. 628 - 634, October 1971.

[Xant99] T. Xanthopoulos, A. P. Chandrakasan, “A Low-Power IDCT Macrocell for MPEG-2 MP@ML Exploiting Data Distribution Properties for Minimal Activity”, IEEE Journal of Solid-State Circuits, vol. 34, no. 5, pp. 693 - 703, May 1999.

[Yang99a] Baoguo Yang, K.B. Letaief, Roger S. Cheng, Zhigang Cao, “Burst Frame Synchronization for OFDM Transmission in Multipath Fading Links”, Proceedings of the IEEE VTC'99, vol.1, pp. 300-304.

[Yang99b] Baoguo Yang, K.B. Letaief, Roger S. Cheng, Zhigang Cao, “An Improved Combined Symbol and Sampling Clock Synchronization Method for OFDM Systems”, Proceedings of the IEEE Wireless Communications and Networking Conference WCNC'99, vol. 3, pp. 1153-1157.